

Операционные системы

Лабораторная работа №2

Буценко В. А.

12 мая 2025

Российский университет дружбы народов, Москва, Россия

4 курс НПМбв-02-21

Информация

Докладчик

- Буценко Варвара Алексеевна
- студент 4 курса НПМбв-02-21
- Российский университет дружбы народов
- 1032200547@pfur.ru
- https://github.com/vabucenko/study_2024-2025_os-intro



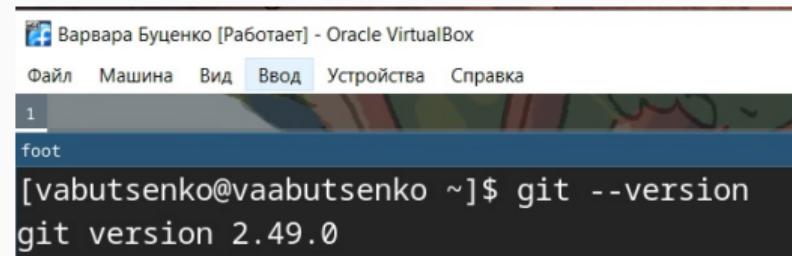
Вводная часть

Задание

- 1) Создать базовую конфигурацию для работы с git.
- 2) Создать ключ SSH.
- 3) Создать ключ PGP.
- 4) Настроить подписи git.
- 5) Зарегистрироваться на Github.
- 6) Создать локальный каталог для выполнения заданий по предмету.

Выполнение лабораторной работы 1

- 1) Устанавливаю git. Заранее сделала это, поэтому использую команду `git --version`, чтобы показать свою версию.



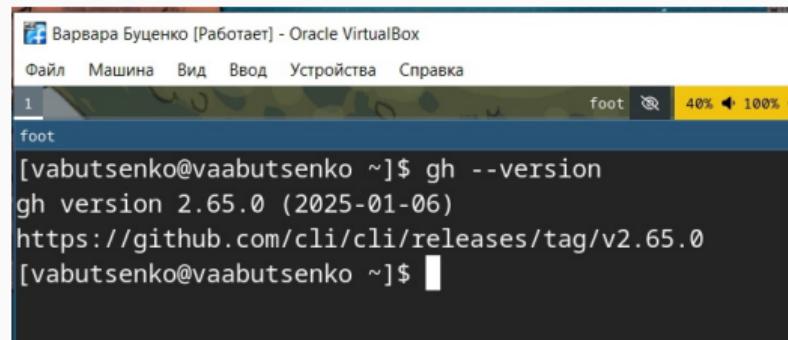
The screenshot shows a terminal window titled "Варвара Буценко [Работает] - Oracle VirtualBox". The window has a menu bar with Russian labels: Файл, Машина, Вид, Ввод (which is highlighted in blue), Устройства, Справка. Below the menu is a toolbar with a single icon labeled "1". The main area of the terminal shows the command line and its output:

```
[vabutsenko@vaabutsenko ~]$ git --version
git version 2.49.0
```

Рис. 1: версия git

Выполнение лабораторной работы 2

2) Устанавливаю gh. Заранее сделала это, поэтому использую команду gh --version, чтобы показать свою версию.



The screenshot shows a terminal window titled "Варвара Буценко [Работает] - Oracle VirtualBox". The window contains the following text:

```
[vabutsenko@vaabutsenko ~]$ gh --version
gh version 2.65.0 (2025-01-06)
https://github.com/cli/cli/releases/tag/v2.65.0
[vabutsenko@vaabutsenko ~]$ █
```

Рис. 2: версия gh

Выполнение лабораторной работы 3

3) Настраиваю git: задаю имя и email владельца:
- git config –global user.name “Name Surname”,
- git config –global user.email “work@mail”.

- Настраиваю utf-8 в выводе сообщений git:
- git config –global core.quotepath false
- Настраиваю верификацию и подписание коммитов git.
- Задаю имя начальной ветки (буду называть её master):
- git config –global init.defaultBranch master
- Параметр autocrlf:
- git config –global core.autocrlf input

Выполнение лабораторной работы 3

- Параметр safecrlf:
- git config --global core.safecrlf warn
- С помощью команды git config --list --show-origin проверяю базовые настройки git.

```
(vabutsenko@vaabutsenko -)$ git config --list --show-origin
file:/home/vabutsenko/.gitconfig      user.name=Varvara Butsenko
file:/home/vabutsenko/.gitconfig      user.email=1032200547@pfur.ru
file:/home/vabutsenko/.gitconfig      core.quotepath=false
file:/home/vabutsenko/.gitconfig      lfs.cachepath=/tmp/git-lfs-cache
file:/home/vabutsenko/.gitconfig      credential.https://github.com.helper=
file:/home/vabutsenko/.gitconfig      credential.https://github.com.helper=/usr/bin/gh auth git-credential
file:/home/vabutsenko/.gitconfig      credential.https://gist.github.com.helper=
file:/home/vabutsenko/.gitconfig      credential.https://gist.github.com.helper=/usr/bin/gh auth git-credential
file:.git/config          core.repositoryformatversion=0
file:.git/config          core.filenode=true
file:.git/config          core.bare=false
file:.git/config          core.logallrefupdates=true
file:.git/config          lfs.repositoryformatversion=0
file:.git/config          lfs.https://github.com/vabutsenko/opera2025.git/info/lfs.locksverify=false
file:.git/config          remote.origin.url=https://github.com/vabutsenko/study_2024-2025_os-intro.git
file:.git/config          remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config          branch.main.remote=origin
file:.git/config          branch.main.merge=refs/heads/main
(vabutsenko@vaabutsenko -)$
```

Выполнение лабораторной работы 4

4) Создаю ключ ssh по алгоритму rsa с ключём размером 4096 бит:

- ssh-keygen -t rsa -b 4096
- Создаю ключ ssh по алгоритму ed25519:
- ssh-keygen -t ed25519
- Проверяю созданные SSH-ключи:
- ls -al ~/.ssh
- cat ~/.ssh/id_ed25519.pub

```
vabutsenko@vabutsenko ~]$ ls -al ~/.ssh
total 16
-rwx----- 1 vabutsenko vabutsenko 180 май  6 19:39 .
-rwx----- 1 vabutsenko vabutsenko 1272 май 14 13:36 ..
-rw----- 1 vabutsenko vabutsenko 444 май  6 15:44 id_ed25519
-rw-r--r-- 1 vabutsenko vabutsenko 121 май  6 15:44 id_ed25519.pub
-rw----- 1 vabutsenko vabutsenko 828 май  6 19:39 known_hosts
-rw-r--r-- 1 vabutsenko vabutsenko 92 май  6 19:39 known_hosts.old
vabutsenko@vabutsenko ~]$ cat ~/.ssh/id_ed25519.pub
sh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJZfUP6BN2G96qqz0Jmf9h0ytz8uwE22lmRpGe9Rd2 Yabutsenko Varvara <1032208547@pfur.ru>
```

Выполнение лабораторной работы 5

5) Генерирую ключ pgp

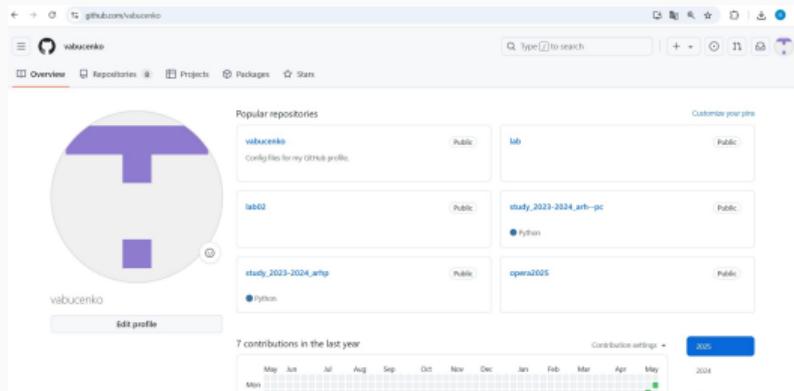
- gpg --full-generate-key Из предложенных опций выбираю:
- тип RSA and RSA;
- размер 4096;
- выбираю срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).
- GPG запросит личную информацию, которая сохранится в ключе:
- Имя: vabucenko
- Адрес электронной почты: 1032200547@pfur.ru

```
[vabutsenko@vaabutsenko ~]$ gpg --list-secret-keys --keyid-format LONG
[keyboxd]
-----
sec   rsa4096/3D14B339EB1DA26B 2025-05-06 [SC]
      D69F196E9B9F0AC8BFF459233D14B339EB1DA26B
uid   [ абсолютна ] varvara <1032200547@pfur.ru>
ssb   rsa4096/478D8F17DC2A3345 2025-05-06 [E]

[vabutsenko@vaabutsenko ~]$ █
```

Выполнение лабораторной работы 6

- 6) Настраиваю github. В прошлом году при прохождении курса “архитектура компьютера” я уже создавала учётную запись, так что использую для выполнения задания именно её.



Выполнение лабораторной работы 7

7) Добавляю PGP ключ в GitHub

- Вывожу список ключей и копирую отпечаток приватного ключа:
- gpg --list-secret-keys --keyid-format LONG
- Перехожу в настройки GitHub (<https://github.com/settings/keys>), нажмаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.

```
[vabutsenko@vaabutsenko ~]$ gpg --list-secret-keys --keyid-format LONG
[keyboxd]
-----
sec   rsa4096/3D14B339EB1DA26B 2025-05-06 [SC]
      D69F196E9B9F0AC8BFF459233D14B339EB1DA26B
uid            [ абсолютно ] varvara <1032200547@pfur.ru>
ssb   rsa4096/478D8F17DC2A3345 2025-05-06 [E]

[vabutsenko@vaabutsenko ~]$
```

The screenshot shows the GitHub user profile page for 'vabutsenko'. On the left, there's a sidebar with account settings like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, and Help & support. The main area displays the 'SSH keys' section under 'Authentication keys'. It shows a single key entry for 'VaButsenko 2025':

Key	Key ID	Added	Last used	Action
	544056...f8ff1e494f1811k2dr5taQ637/C88igf9OefnWQ	Added on 6 May 2025	—	Delete

Below the table, there's a note: 'Check out our guide to [connecting to GitHub using SSH keys](#) or [troubleshoot common SSH problems](#)'.

Выполнение лабораторной работы 8

8) Проверяю подписи коммитов

- git config --global user.signingkey
- git config --global commit.gpgsign
- git log --show-signature -1

```
[vabutsenko@vaabutsenko ~]$ git config --global user.signingkey
[vabutsenko@vaabutsenko ~]$ git config --global commit.gpgsign
[vabutsenko@vaabutsenko ~]$ git log --show-signature -1
commit 71becdf27777ee72f49c92752eacf9aec10a31d6 (HEAD -> main, origin/main)
Author: Varvara Butsenko <1032200547@pfur.ru>
Date:   Thu May 8 02:58:31 2025 +0300

fix: remove large files and update .gitignore
```

9) Проверяю аутентификации в GitHub CLI

- gh auth status
- Создаю шаблон рабочего пространства.
- mkdir -p ~/work/study/2024-2025/“Operacionnie systems”
- cd ~/work/study/2024-2025/“Operacionnie systems”
- gh repo create study_2024-2025_os-intro
- –template=yamadharma/course-directory-student-template –publicgit clone –recursive
git@github.com:/study_2024-2025_os-intro.git os-intro

Выполнение лабораторной работы 9

- Перехожу в каталог курса:
- cd ~/work/study/2024-2025/“Operacionnie systems”/os-intro
- Удаляю лишние файлы:
- rm package.json
- Создаю необходимые каталоги:
- echo os-intro > COURSE
- make

Выполнение лабораторной работы 9

- Отправляю файлы на сервер:
- `git add .`
- `git commit -am 'feat(main): make course structure'`
- `git push`

```
[vabutsenko@vaabutsenko ~]$ gh auth status
github.com
  ✓ Logged in to github.com account vabucenko (keyring)
  - Active account: true
  - Git operations protocol: https
  - Token: gho_*****
  - Token scopes: 'gist', 'read:org', 'repo', 'workflow'
[vabutsenko@vaabutsenko ~]$ █
```

Выполнение лабораторной работы 9

```
[vabutsenko@vaabutsenko work]$ tree -L 3
.
└── rus_fixed_full_fixed.map
    └── study
        └── 2024-2025
            └── Operacionnie systems

4 directories, 1 file
[vabutsenko@vaabutsenko work]$
```

The screenshot shows a GitHub repository page for 'vabutsenko/study_2024-2025_os-intro/tree/master'. The commit history lists 15 commits, all labeled 'Initial commit' or 'Добавлен отчёт для лабы'. The commits are dated from 17 hours ago to 2 days ago. The repository has 5 contributors. On the right side, there are sections for 'Оптимизация, лейб-стайла или темы не предложено', 'Проект меня', 'Лицензия CC-BY-4.0', 'Листинги', 'Файл', 'Темы', 'Языки', and 'Предлагаемые рабочие процессы'.

Commit Message	Date	Author
Добавлен отчёт для лабы	17 hours ago	vabutsenko
Initial commit	17 hours ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	1 hour ago	vabutsenko
Добавлен отчёт для лабы	1 hour ago	vabutsenko
Initial commit	2 days ago	vabutsenko

Выполнение лабораторной работы 9

```
[vabutsenko@vaabutsenko work]$ git remote -v
origin  git@github.com:vabucenko/study_2024-2025_os-intro.git (fetch)
origin  git@github.com:vabucenko/study_2024-2025_os-intro.git (push)
[vabutsenko@vaabutsenko work]$ git branch -vv
* main 71becdf [origin/main] fix: remove large files and update .gitignore
```

Контрольные вопросы 1

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

Системы контроля версий (VCS) — это программные инструменты для отслеживания изменений в файлах (чаще всего в исходном коде) и координации работы нескольких участников. Они позволяют фиксировать историю изменений, возвращаться к предыдущим версиям и совместно работать над проектами.

Для решения каких задач предназначаются:

- Хранение истории изменений
- Командная разработка
- Резервное копирование
- Анализ изменений

- 2) Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
- Хранилище - база данных, хранящая всю историю проекта (файлы, изменения, авторов).
 - Коммит - снимок состояния файлов на определённый момент времени.
 - История - цепочка коммитов, отражающая эволюцию проекта.
 - Рабочая копия - текущие файлы в вашей папке проекта, с которыми вы работаете.

- 3) Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
- Ключевое различие — в способе хранения истории изменений и организации работы.
 - Централизованные это системы, где вся история проекта хранится на едином сервере. Разработчики получают из него последние версии файлов и отправляют изменения обратно.
 - Децентрализованные - это системы, где каждый участник проекта имеет полную копию репозитория со всей историей изменений. Работа ведётся локально, а синхронизация между копиями происходит через команды push/pull.

Контрольные вопросы 3

Централизованные: - SVN - CVS - Perforce

Децентрализованные: - Git - Mercurial - Bazaar

- 4) Опишите действия с VCS при единоличной работе с хранилищем.
- Инициализация репозитория
 - Ежедневные действия (проверить изменения, добавить файлы в индекс, зафиксировать изменения)
 - Просмотр истории
 - Работа с ветками
 - Синхронизация с удалённым репозиторием
 - Отмена изменений (при ошибках)

- 5) Опишите порядок работы с общим хранилищем VCS.
- Клонирование репозитория
 - Создание своей ветки
 - Ежедневная работа (внесение изменений в файлы, регулярное сохранение изменений (коммиты))
 - Синхронизация с общим репозиторием
 - Создание Pull/Merge Request
 - Рецензирование и исправления
 - Слияние изменений
 - Удаление отработанной ветки

Контрольные вопросы 6

6) Каковы основные задачи, решаемые инструментальным средством git?

- Контроль версий
- Командная разработка
- Ветвление и слияние
- Резервное копирование
- Отслеживание изменений
- Тестирование идей
- Автоматизация процессов
- Распределённая разработка

7) Назовите и дайте краткую характеристику командам git.

- Настройка:
 - git config Настройка параметров Git (имя пользователя, email, редактор и др.).
 - git config -global user.name “Ваше Имя” Создание и клонирование репозиториев
 - git init Создает новый локальный репозиторий в текущей папке.
 - git clone Клонирует удаленный репозиторий на локальную машину.
- Работа с изменениями:
 - git status Показывает состояние файлов (измененные, добавленные, неотслеживаемые).
 - git add Добавляет файлы в индекс (staging area) для последующего коммита.
 - git commit -m “сообщение” Фиксирует изменения в репозитории с комментарием

Контрольные вопросы 7

- Просмотр истории:
- `git log` Выводит историю коммитов (автор, дата, сообщение).
- `git diff` Показывает разницу между текущими изменениями и последним коммитом. Ветвление и слияние
- `git branch` Показывает список веток (текущая помечена *).
- `git checkout` Переключается на указанную ветку.
- `git merge` Влияет изменения из указанной ветки в текущую.
- Работа с удаленными репозиториями
- `git remote add` Добавляет удаленный репозиторий (например, `origin`).
- `git push` Отправляет локальные изменения на удаленный сервер.

- Отмена изменений:
- `git reset –hard HEAD` Сбрасывает все изменения до последнего коммита (осторожно!).
- `git revert` Создает новый коммит, отменяющий указанный.

- 8) Приведите примеры использования при работе с локальным и удалённым репозиториями.
- Работа с локальными репозиториями
 - Создание локального репозитория: `git init my_project`
 - Клонирование удалённого репозитория в локальный: `git clone https://github.com/username/repo.git`
 - Добавление файлов в локальный репозиторий: `git add filename.txt`
 - Коммит изменений в локальном репозитории: `git commit -m "Добавлен новый файл"`
 - Просмотр статуса локального репозитория:`git status`
 - Работа с удалёнными репозиториями

Контрольные вопросы 9

9) Что такое и зачем могут быть нужны ветви (branches)?

- Ветвь — это независимая линия разработки, позволяющая изолировать изменения от основного кода (обычно main/master). Каждая ветка содержит свою историю коммитов.
- Зачем нужны ветки?
- Параллельная разработка
- Изоляция экспериментов
- Гибкое управление версиями
- Контроль качества
- Упрощение Code Review

10) Как и зачем можно игнорировать некоторые файлы при commit?

- Чтобы игнорировать файлы в Git, используется файл `.gitignore`, который находится в корневой директории репозитория. Этот файл содержит шаблоны, которые указывают, какие файлы или директории должны быть проигнорированы.
- Зачем игнорировать файлы?
- Не коммитить временные/системные файлы Логи, кэш, бинарники (например, `.log`, `.tmp`, `.exe`).
- Избегать личных настроек Файлы IDE (`.idea/`, `.vscode/`), конфиги с паролями.
- Снижать «шум» в репозитории Автогенерируемые файлы (`node_modules/`, `pycache/`).

Выводы

- В ходе выполнения лабораторной работы были успешно выполнены все поставленные задачи, связанные с изучением идеологии и применением средств контроля версий, а также освоением навыков работы с системой Git. В процессе работы были достигнуты следующие результаты:
- 1) Настройка базовой конфигурации Git: Были выполнены основные настройки Git, включая указание имени пользователя, email, настройку кодировки UTF-8, а также параметров для работы с ветками и окончаниями строк. Это обеспечило корректную работу системы контроля версий.
 - 2) Создание и настройка SSH и PGP ключей: Были сгенерированы ключи SSH (алгоритмы RSA и ed25519) и PGP, что позволило обеспечить безопасное взаимодействие с удалёнными репозиториями и подписывание коммитов. Ключи были успешно добавлены в аккаунт GitHub.

- 3) Работа с GitHub: Была выполнена аутентификация в GitHub CLI, создан шаблон рабочего пространства, а также организована структура каталога для выполнения заданий. Локальные изменения были зафиксированы и отправлены на удалённый репозиторий.
- 4) Освоение команд Git: В процессе работы были изучены и применены основные команды Git, такие как git init, git clone, git add, git commit, git push, git pull, а также команды для работы с ветками и историей изменений. Это позволило эффективно управлять версиями проекта.