

# **Отчёт по лабораторной работе № 2**

**Операционные системы**

Диого Элизеу Луиж Музумбо

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Установка программного обеспечения . . . . .	6
3.2	Базовая настройка git . . . . .	7
3.3	Создали ключи ssh . . . . .	7
3.4	Создали ключи pgr . . . . .	8
3.5	Настройка github . . . . .	9
3.6	Добавление PGP ключа в GitHub . . . . .	10
3.7	Настройка автоматических подписей коммитов git . . . . .	12
3.8	Настройка gh . . . . .	12
3.9	Создание репозитория курса на основе шаблона . . . . .	13
3.10	Настройка каталога курса . . . . .	14
<b>4</b>	<b>Выводы</b>	<b>16</b>
<b>5</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

# Список иллюстраций

3.1	6
3.2	6
3.3	7
3.4	7
3.5	7
3.6	7
3.7	7
3.8	8
3.9	8
3.10	9
3.11	9
3.12	10
3.13	10
3.14	11
3.15	11
3.16	11
3.17	12
3.18	12
3.19	12
3.20	13
3.21	13
3.22	14
3.23	14
3.24	14
3.25	14
3.26	15
3.27	15
5.1	18

# 1 Цель работы

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

## 2 Задание

- Установить и настроить ПО для работы с git.

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

Установили git:(рис. [3.1])

```
[root@fedora ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:21:16 назад, Вс 25 июн
2023 10:12:57.
Пакет git-2.35.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 3.1: .

Установили gh:(рис. [3.2])

```
[root@fedora ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:22:52 назад, Вс 25 июн
2023 10:12:57.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.23.0-1.fc36  updates      8.2 М
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]: д
Загрузка пакетов:
gh-2.23.0-1.fc36.x86_64.rpm                2.7 MB/s | 8.2 MB    00:03
-----
Общий размер                               2.3 MB/s | 8.2 MB    00:03
```

Рис. 3.2: .

## 3.2 Базовая настройка git

Задали имя и email владельца репозитория: (рис. [3.3])

```
[elmdiago@fedora ~]$ git config --global user.name "EliseuDiogo"
[elmdiago@fedora ~]$ git config --global user.email "<muzumbodiogo2@gmail.com>"
[elmdiago@fedora ~]$
```

Рис. 3.3: .

Настроили utf-8 в выводе сообщений git:(рис. [3.4])

```
[elmdiago@fedora ~]$ git config --global core.quotepath false
```

Рис. 3.4: .

Настроили верификацию и подписание коммитов git. Задали имя начальной ветки (будем называть её master).(рис. [3.5])

```
[elmdiago@fedora ~]$ git config --global init.defaultBranch master
```

Рис. 3.5: .

Параметр autocrlf:(рис. [3.6])

```
[elmdiago@fedora ~]$ git config --global core.autocrlf input
```

Рис. 3.6: .

Параметр safecrlf: (рис. [3.7])

```
[elmdiago@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 3.7: .

## 3.3 Создали ключи ssh

по алгоритму rsa с ключём размером 4096 бит: (рис. [3.8])

```
[elmdigo@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/elmdigo/.ssh/id_rsa):
/home/elmdigo/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/elmdigo/.ssh/id_rsa
Your public key has been saved in /home/elmdigo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:36XbYIKjdWUi+R2eyZLqf7oaD4im/7wvp8qvMsKltdM elmdigo@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|
|             .
|          S . + .
|       o . . = X *
|  . + = . * B @
| .oo* E.o.B +.+
| ..+*=BB=0=+. .
+---[SHA256]-----+
```

Рис. 3.8: .

по алгоритму ed25519: (рис. [3.9])

```
+---[SHA256]-----+
[elmdigo@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/elmdigo/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/elmdigo/.ssh/id_ed25519
Your public key has been saved in /home/elmdigo/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:EZKzvq9WGFtLZQgiS4u44j0liYniuY2YyeAlZPaMONG elmdigo@fedora
The key's randomart image is:
+--[ED25519 256]--+
|   o o.o. oo   |
| . o +oo .+.   |
| . . o o.o     |
| .   o ..      |
|o+ . . =S      |
|@oB  + .       |
|XOE+  o        |
|B+O   o        |
|+B.. ..o.      |
+---[SHA256]-----+
```

Рис. 3.9: .

### 3.4 Создали ключи рдр

Сгенерировали ключ (рис. [3.10])



Из предложенных опций выбирали: тип RSA and RSA; размер 4096; выбрали срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). GPG запросил личную информацию, которая сохранится в ключе: Имя. Адрес электронной почты. При вводе email убедились, что он соответствует адресу, используемому на GitHub. (рис. [3.11])

```
+-----[SHA256]-----+
[elmdio@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; Copyright (C) 2021 Free Software Foundation, Inc.
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/elmdio/.gnupg'
gpg: создан шит с ключами '/home/elmdio/.gnupg/pubring.kbx'
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
```

Рис. 3.10: .

```
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2023-06-25 [SC]
     B54AE21C9F3F344621243053F7773513E44B4D4A
uid                               Eliseu Diogo <muzumbodiogo2@gmail.com>
sub   rsa4096 2023-06-25 [E]
```

Рис. 3.11: .

## 3.5 Настройка github

Создайте учётную запись на github.com. (рис. [3.12])

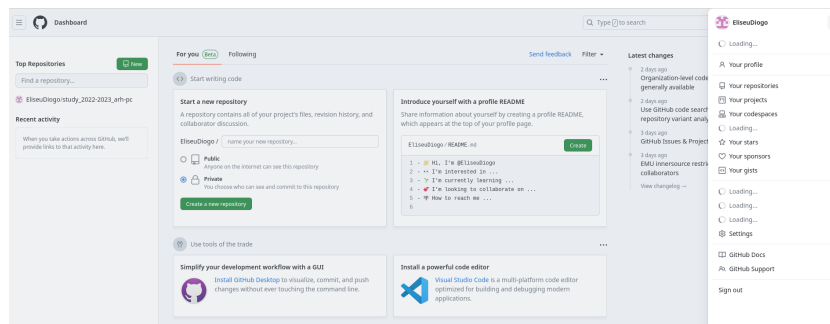


Рис. 3.12: .

Заполните основные данные на github.com. (рис. [3.13])

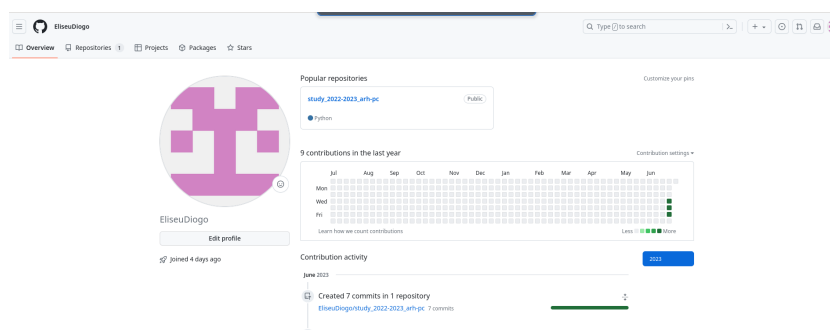


Рис. 3.13: .

## 3.6 Добавление PGP ключа в GitHub

Вывели список ключей и копировали отпечаток приватного ключа: (рис. [3.14])  
Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

```
[elmdio@fedora ~]$ 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
/home/elmdio/.gnupg/pubring.kbx
-----
sec   rsa4096/F7773513E44B4D4A 2023-06-25 [SC]
      B54AE21C9F3F344621243053F7773513E44B4D4A
uid           [ абсолютно ] Eliseu Diogo <muzumbodiogo2@gmail.com>
ssb   rsa4096/9540A40B23695250 2023-06-25 [E]
[elmdio@fedora ~]$
```

Рис. 3.14: .

Скопировали сгенерированный PGP ключ в буфер обмена: (рис. [3.15])

```
[elmdio@fedora ~]$ gpg --armor --export | xclip -sel clip
```

Рис. 3.15: .

Перешли в настройки GitHub, нажали на кнопку New GPG key и вставили полученный ключ в поле ввода. (рис. [3.16], [3.17])

#### GPG keys / Add new

Title

Key

```
R95MnNWSIKV0p6h/EmEQgZnZn0i9/Dg0+NA1LgSaLHM4q5EGpyt2oC1P1gm1ZYJM
2Swq65EZLxBdBUru5BKb6DKHYvpL5Jn7ughCveO2FcTEFSPXKSX/TD8HqW4aiG6n
ek0xSxpth9qfSGS3QgU4kslCptm4wYi7Pp0ht6qGg7yjbZfXbyIrX57Qv3qgB4HL
qyjTe08SaUiis3/wCqdRoD3OexhTq6dhNksoklFWbmPYiN9DbpozUZ7BY6NWN73b
Oba4YcDhbgRbF7ZyBpSTRrAYvYuZhiQ3c1Y+MUysWL4p9meKniHWNTL7NgIE/Ejc
zL0ertTjNjHREmdInLkCDQRkl/xKARAA3p4AAnnBhEGhIIvUAN9ja/XVEgfbijfr
PVQJW+IJQL9HuqAQHzBVjNyxRO1Jbu/FO2E0ojs4jelN3YgDiP/FOyZ8a/ojHI7
cxY+6rOu/SRzMPjADfgnGkyeCvCyHj4suGYAY5/2DhednTKXF/hLH2AQR+Bj9jD1
Lx14SG/MYImWD4xtYiTB6jpAmFUQxM9dfHv9pwB9on7TDMvTGemNBjDctlrbaIX
```

Add GPG key

Рис. 3.16: .

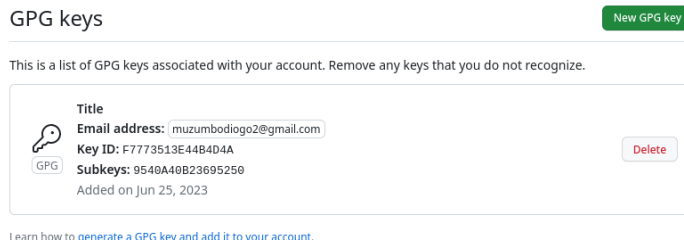


Рис. 3.17: .

## 3.7 Настройка автоматических подписей коммитов git

Используя введённый email, указали Git применять его при подписи коммитов: (рис. [3.18])

```
[elmdiago@fedora ~]$ gpg --armor --export | xclip -sel clip
[elmdiago@fedora ~]$ git config --global user.signingkey muzumbodiogo2@gmail.com
[elmdiago@fedora ~]$ git config --global commit.gpgsign true
[elmdiago@fedora ~]$ git config --global gpg.program $(which gpg2)
[elmdiago@fedora ~]$
```

Рис. 3.18: .

## 3.8 Настройка gh

Авторизовались в gh. (рис. [3.19]) Утилита задали несколько наводящих вопросов.

```
[elmdiago@fedora ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/elmdiago/.ssh/id_rsa.
pub
? Title for your SSH key: Title
? How would you like to authenticate GitHub CLI? Login with a web browser
! First copy your one-time code: 8087-9F2F
Press Enter to open github.com in your browser...
```

Рис. 3.19: .

## 3.9 Создание репозитория курса на основе шаблона

Создали шаблон рабочего пространства. (рис. [3.20], [3.21], [3.22])

```
[elmdiago@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[elmdiago@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[elmdiago@fedora Операционные системы]$
```

Рис. 3.20: .

The screenshot shows the GitHub web interface for creating a new repository from a template. The title is 'Create a new repository from course-directory-student-template'. Below the title, it says 'The new repository will start with the same files and folders as [yamadharm/course-directory-student-template](#).' The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'EliseuDiogo' with a dropdown arrow. The 'Repository name' is 'study\_2022-2023\_os-intro' with a green checkmark. Below this, a message says 'Great repository names are [study\\_2022-2023\\_os-intro](#) is available. ration? How about [reimagined-octo-fishstick](#)?'. There is a 'Description (optional)' text area. Below that, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.' Below the visibility options, there is a checkbox for 'Include all branches' with a subtext: 'Copy all branches from yamadharm/course-directory-student-template and not just master.' At the bottom, there is a note: 'You are creating a public repository in your personal account.' and a green button labeled 'Create repository from template'.

Рис. 3.21: .

```
[elmdiego@fedora Операционные системы]$ git clone --recursive git@github.com:EliseuDiogo/study_2022-2023_os-intro.git
Клонирование в «study_2022-2023_os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 8.47 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/elmdiego/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 849.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/elmdiego/work/study/2022-2023/Операционные системы/study_2
```

Рис. 3.22: .

## 3.10 Настройка каталога курса

Перешли в каталог курса: (рис. [3.23])

```
[elmdiego@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/study_2022-2023_os-intro
[elmdiego@fedora study_2022-2023_os-intro]$
```

Рис. 3.23: .

Удалили лишние файлы: (рис. [3.24])

```
[elmdiego@fedora study_2022-2023_os-intro]$ rm package.json
[elmdiego@fedora study_2022-2023_os-intro]$
```

Рис. 3.24: .

Создали необходимые каталоги: (рис. [3.25])

```
[elmdiego@fedora study_2022-2023_os-intro]$ echo os-intro > COURSE
[elmdiego@fedora study_2022-2023_os-intro]$ make
[elmdiego@fedora study_2022-2023_os-intro]$
```

Рис. 3.25: .

Отправили файлы на сервер: (рис. [3.26], [3.27])

```
[elmdiago@fedora study_2022-2023_os-intro]$ git add .
[elmdiago@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
[master f162f57] feat(main): make course structure
361 files changed, 100327 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric-csl
```

Рис. 3.26: .

```
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
[elmdiago@fedora study_2022-2023_os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 343.07 КиБ | 2.40 МиБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:EliseuDiogo/study_2022-2023_os-intro.git
22d42ae..f162f57 master -> master
[elmdiago@fedora study_2022-2023_os-intro]$
```

Рис. 3.27: .

## **4 Выводы**

В ходе выполнения данной лабораторной работы была изучена идеология и применение средств контроля версий и освоены умения по работе с git.



## 5 Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система управления версиями (также используется определение «система контроля версий», от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации. Версия (revision), или ревизия, — состояние всего хранилища или отдельных файлов в момент времени («пункт истории»). Commit («трудовой вклад», не переводится) — процесс создания новой версии; иногда синоним версии. Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Децентрализованные VCS: У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория (Git, Mercurial, Bazaar).

Централизованные VCS : Одно основное хранилище всего проекта Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно (Subversion, CVS, TFS, VAULT, AccuRev)

6. Каковы основные задачи, решаемые инструментальным средством git? У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.
7. Назовите и дайте краткую характеристику командам git. git init - создание репозитория git add (имена файлов) - Добавляет файлы в индекс git commit – выполняет коммит проиндексированных файлов в репозиторий git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы git checkout (sha1 или метка) - получение указанной версии файла git push – отправка изменений в удаленный репозиторий git fetch – получение изменений из удаленного репозитория git clone (remote url) - клонирование удаленного репозитория себе
8. Приведите примеры использования при работе с локальным и удалённым репозиториями. (рис. [5.1])

```
[elmdio@fedora study_2022-2023_os-intro]$ git add .
[elmdio@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
[master f162f57] feat(main): make course structure
361 files changed, 100327 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
```

Рис. 5.1: .

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

Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала Основная ветка– master Ветки в GIT. Показать все ветки, существующие в репозитории git branch. Создать ветку git branch имя.

Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

## 10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. Вот некоторые распространенные примеры таких файлов:

кэши зависимостей, например содержимое node\_modules или packages; скомпилированный код, например файлы .o, .рус и .class ; каталоги для выходных данных сборки, например bin, out или target; файлы, сгенерированные во время выполнения, например .log, .lock или .tmp; скрытые системные файлы, например .DS\_Store или Thumbs.db; личные файлы конфигурации IDE, например .idea.workspace.xml.