

# Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютеров и операционных систем

Овезов Мерген

## Содержание

1	Цель работы.....	1
2	Задание .....	1
3	Выполнение лабораторной работы.....	3
3.1	Установка программного обеспечения .....	3
3.2	Базовая настройка git .....	3
3.3	Создание ключа SSH.....	4
3.4	Создание Ключа GPG .....	5
3.5	Регистрация на GitHub .....	5
3.6	Добавление ключа GPG в GitHub .....	6
3.7	Настроить подписи Git .....	6
3.8	Настройка gh .....	7
3.9	Создание репозитория курса на основе шаблона .....	7
3.10	Выводы .....	8

## 1 Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

## 2 Задание

1.Создать базовую конфигурацию для работы с git 2.Создать ключ SSH 3.Создать ключ GPG 4.Настроить подписи Git 5.Зарегистрироваться на GitHub 6.Создать локальный каталог для выполнения заданий по предмету # Теоретическое введение

Основные команды git

Перечислим наиболее часто используемые команды git.

Создание основного дерева репозитория:

git init

Получение обновлений (изменений) текущего дерева из центрального репозитория:

```
git pull
```

Отправка всех произведённых изменений локального дерева в центральный репозиторий:

```
git push
```

Просмотр списка изменённых файлов в текущей директории:

```
git status
```

Просмотр текущих изменений:

```
git diff
```

Сохранение текущих изменений:

добавить все изменённые и/или созданные файлы и/или каталоги:

```
git add .
```

добавить конкретные изменённые и/или созданные файлы и/или каталоги:

```
git add имена_файлов
```

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

```
git rm имена_файлов
```

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы:

```
git commit -am 'Описание коммита'
```

сохранить добавленные изменения с внесением комментария через встроенный редактор:

```
git commit новой ветки, базирующейся на текущей:
```

```
git checkout -b имя_ветки
```

переключение на некоторую ветку:

```
git checkout имя_ветки
```

(при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий:

```
git push origin имя_ветки
```

слияние ветки с текущим деревом:

`git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки:

`git branch -d имя_ветки`

принудительное удаление локальной ветки:

`git branch -D имя_ветки`

удаление ветки с центрального репозитория:

`git push origin :имя_ветки`

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

Установка необходимого программного обеспечения git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh` (рис.1).

```
root@sparrow:~# dnf install git
Last metadata expiration check: 0:05:49 ago on Sat 22 Jun 2024 06:53:35 AM EDT.
Package git-2.44.0-1.fc40.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

Рис 1

```
root@sparrow:~# dnf install gh
Last metadata expiration check: 0:06:36 ago on Sat 22 Jun 2024 06:53:35 AM EDT.
Dependencies resolved.
=====
Package      Architecture Version      Repository    Size
=====
Installing:
gh           x86_64      2.45.0-1.fc40 fedora        8.7 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 8.7 M
Installed size: 43 M
Is this ok [y/N]: y
Downloading Packages:
gh-2.45.0-1.fc40.x86_64.rpm                5.9 MB/s | 8.7 MB  00:01
-----
Total                                       3.0 MB/s | 8.7 MB  00:02
Running transaction check
Transaction check succeeded.
Running transaction test
```

Git

Рис 1

### 3.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту (рис 2)

```
root@sparrow:~# git config --global user.name "Mergen Ovezov"
root@sparrow:~# git config --global user.email "1032234249@pfur.ru"
```

Рис 2

Настраиваю utf-8 в выводе сообщений git для их корректного отображения

```
root@sparrow: # git config --global core.quotepath false
```

Рис 3

Начальной ветке задаю имя master

```
root@sparrow: # git config --global init.defaultBranch master
```

Рис 4

Задаю параметры autocrlf и safecrlf для корректного отображения конца строки

```
root@sparrow: # git config --global core.autocrlf input
root@sparrow: # git config --global core.safecrlf warn
```

Рис 5

### 3.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa

```
root@sparrow: # ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (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:z0wg/XCTNt/UQnWNZ4IzJo2H6y2StCswzk28DdA/rjk root@sparrow
The key's randomart image is:
+---[RSA 4096]-----+
|          + ..O+ |
|         . +.B.O.= |
|        .. + *= oo+. |
|       . ...*..+ O . |
|      o oS+o.. . |
|     o + *=o . |
|    o = = ++. |
|   o E.+ |
|  o+ |
+-----[SHA256]-----+
```

Рис 6

Создаю ключ ssh по алгоритму ed 25519

```

root@sparrow:~# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (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:e1RlKKpNt/oxpiAUkufJFIV2x6ilgh7grhx8x0mlpSU root@sparrow
The key's randomart image is:
+--[ED25519 256]--+
| oE+*.o      .o |
| .. =0.+ o . .o |
| o.+==+ o . . |
| oo*oo. o . |
| ....= . +S... |
| o.. . .O. |
| o . . . . = |
| .. . . .+ o |
| ... |
+-----[SHA256]-----+
1

```

Рис 7

### 3.4 Создание Ключа GPG

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю макс. длину ключа, 4096, оставляю неограниченный срок действия ключа. Далее отвечаю на вопросы программы о личной информации

```

(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.
Real name: OvezovMergen

```

Рис 8

### 3.5 Регистрация на GitHub

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполнил и проводил его настройку, поэтому просто вхожу в свой аккаунт

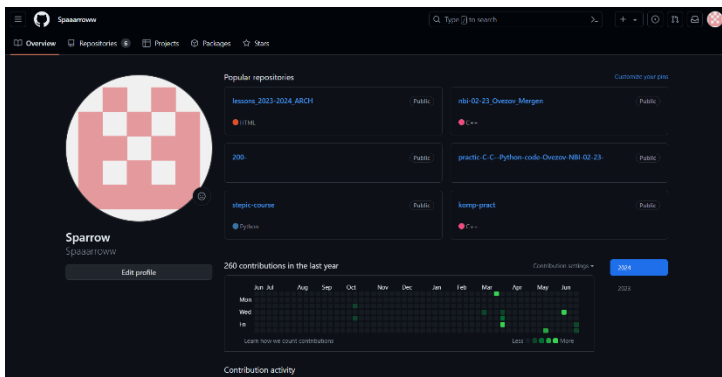


Рис 9

### 3.6 Добавление ключа GPG в GitHub

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа (последовательность байтов для идентификации более длинного, по сравнению с самим отпечатком, ключа), он стоит после знака слеша, копирую его в буфер обмена

Рис 10

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip.

```
root@10: # gpg --armor --export 258ECFD6A5F8CC24 | xclip -sel clip
```

Рис 11

Открываю настройки GitHub, ищу среди них добавление GPG ключа. Нажимаю на New GPG key и вставляю в поле ключ из буфера обмена. Я добавил ключ GPG на GitHub

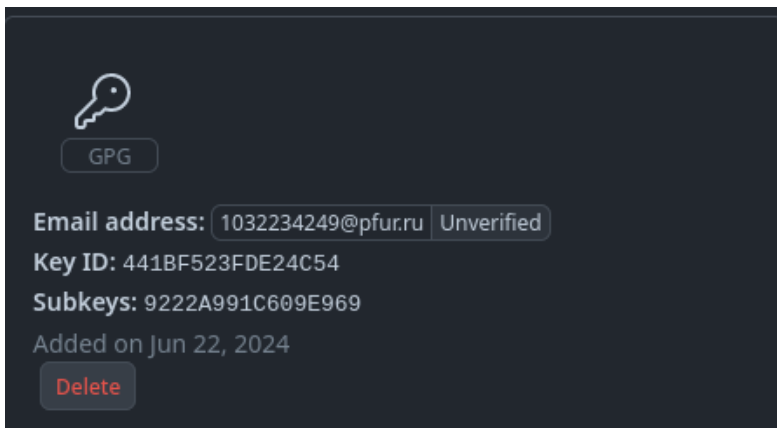


Рис 12

### 3.7 Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов.

```
root@sparrow:~# gpg --armor --export 441BF523FDE24C54 | xclip -sel clip
root@sparrow:~# git config --global user.signingkey 441BF523FDE24C54
root@sparrow:~# git config --global commit.gpgsign true
root@sparrow:~# git config --global gpg.program $(which gpg)
```

Рис 13

### 3.8 Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер

```
root@sparrow:~# gh auth login
? What account do you want to log into? 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: DD55-C977
Press Enter to open github.com in your browser... █
```

Рис 14

Завершаю авторизацию на сайте

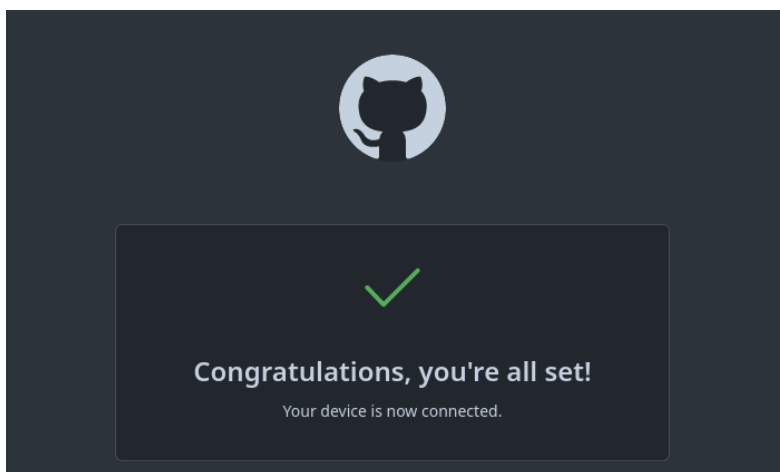


Рис 15

Вижу сообщение о завершении авторизации под именем aogutyansev

```
Press Enter to open github.com in your browser...
! Authentication complete.
- gh config set -h github.com git_protocol https
Configured git protocol
Authentication credentials saved in plain text
Logged in as Mergen24
root@sparrow:~#
```

Рис 16

### 3.9 Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты mkdir и флага -p, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты cd перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду `gh repo create study_2023-2024_os-intro -template yamadharma/course-derctory-student-trmplate-public`, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом https, а не ssh, потому что при авторизации в gh выбрал протокол https

```
liveuser@sparrow:~$ git clone --recursive https://github.com/Mergen24/study_2023-2024.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
КБ Receiving objects: 100% (3/3), done.
```

Рис 17

Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls`

```
/2023-2024/Операционные системы$ cd os-intro
/2023-2024/Операционные системы/os-intro$
```

Рис 18

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile`

```
~/work/study/2023-2024/Операционные системы/os-intro$ rm package.
~/work/study/2023-2024/Операционные системы/os-intro$ echo os-intro
os-intro
~/work/study/2023-2024/Операционные системы/os-intro$ make
```

Рис 19

Добавляю все новые файлы для отправки на сервер(сохраняю добавленные изменения с помощью команды `git add` и комментирую их с помощью `git commit`

```
~/work/study/2023-2024/Операционные системы/os-intro$ git add .
~/work/study/2023-2024/Операционные системы/os-intro$ git commit
```

Рис 20

Отправляю файлы на сервер с помощью `git push`.

```
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 281 байт | 281.00 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 https://github.com/aorumyancev/study_2023-2024_os-intro.git
6ad0403..68f63e4 master -> master
```

Рис 21

### 3.10 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, освоил умение по работе с `git`.