

# **Отчет по лабораторной работе №2**

**Дисциплина архитектура компьютера**

Никуленков Степан Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>

# Список иллюстраций

4.1	установка . . . . .	8
4.2	установка . . . . .	8
4.3	базовая настройка git . . . . .	9
4.4	создание ключей . . . . .	10
4.5	окно подтверждения . . . . .	10
4.6	ключ . . . . .	11
4.7	копирование ключа . . . . .	11
4.8	добавление ключа . . . . .	12
4.9	создание репозитория . . . . .	12
4.10	отправка на сервер . . . . .	13

# 1 Цель работы

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

## 2 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.

## 3 Теоретическое введение

Системы контроля версий. Общие понятия

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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными

участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

## 4 Выполнение лабораторной работы

Устанавливаю git: `dnf install git`

```
Cloning into 'study_2024-2025_os-intro'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Рис. 4.1: установка

Устанавливаю gh: `dnf install gh`

```
Fedora 40 - x86_64
Fedora 40 openh264 (from Cisco) - x86_64
Fedora 40 - x86_64 - Updates
Fedora 40 - x86_64 - Updates
google-chrome
google-chrome
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Driver
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Driver
RPM Fusion for Fedora 40 - Nonfree - Steam
RPM Fusion for Fedora 40 - Nonfree - Steam
Package git-2.44.0-1.fc40.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
root@server:~# dnf install gh
Last metadata expiration check: 0:00:16 ago on Sat 08 Mar 2025 03:26:58 PM MSK.
Dependencies resolved.
=====
Package                               Architecture      Version
-----
Installing:                             x86_64            2.65.0-1.fc40
gh
=====
Transaction Summary
-----
Install 1 Package

Total download size: 11 M
Installed size: 50 M
Is this ok [y/N]: y
Downloading Packages:
gh-2.65.0-1.fc40.x86_64.rpm
-----
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : 
  Installing     : gh-2.65.0-1.fc40.x86_64
  Running scriptlet: gh-2.65.0-1.fc40.x86_64
Installed:
gh-2.65.0-1.fc40.x86_64
```

Рис. 4.2: установка

Базовая настройка git

Задаю имя и email для моего репозитория:

```
git config --global user.name "Stepan Nikulenkov" git config --global user.email
"stepannikulenkov@gmail.com"
```



Настраиваю utf-8 в выводе сообщений git:

```
git config --global core.quotePath false
```

Настраиваю верификацию и подписание коммитов git

Задаю имя начальной ветки (будем называть её master):

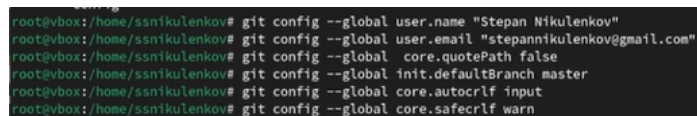
```
git config --global init.defaultBranch master
```

Параметр autocrlf:

```
git config --global core.autocrlf input
```

Параметр safecrlf:

```
git config --global core.safecrlf warn
```

A screenshot of a terminal window showing a series of git configuration commands being executed. The prompt is 'root@vbox:/home/ssnikulenkov#'. The commands are: 'git config --global user.name "Stepan Nikulenkov"', 'git config --global user.email "stepannikulenkov@gmail.com"', 'git config --global core.quotePath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'.

```
root@vbox:/home/ssnikulenkov# git config --global user.name "Stepan Nikulenkov"
root@vbox:/home/ssnikulenkov# git config --global user.email "stepannikulenkov@gmail.com"
root@vbox:/home/ssnikulenkov# git config --global core.quotePath false
root@vbox:/home/ssnikulenkov# git config --global init.defaultBranch master
root@vbox:/home/ssnikulenkov# git config --global core.autocrlf input
root@vbox:/home/ssnikulenkov# git config --global core.safecrlf warn
```

Рис. 4.3: базовая настройка git

Создаю ключи ssh по алгоритму rsa с ключём размером 4096 бит:

```
ssh-keygen -t rsa -b 4096
```

по алгоритму ed25519:

```
ssh-keygen -t ed25519
```

Создаю ключи pgp

Генерируем ключ gpg --full-generate-key

```
-----[RSA 4096]-----+
| .++      o o. |
| =+      . o o |
| o.o.      + |
| +.o      o |
| +.o+ S. . . |
| o o+o+ o . |
| . =o.o.o. . |
| E+=o.+o . |
| .BB...+.. |
|-----[SHA256]-----+
root@vbox:/home/ssnikulenkova# gpg --full-generate-key
gpg (GnuPG) 2.4.4; 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: directory '/root/.gnupg' created
Please select what kind of key you want:
  (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 keysizes do you want? (3072) 4096
Requested keysizes is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <nw> = key expires in n weeks
  <nm> = key expires in n months
  <ny> = 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: Stepan
Email address: stepannikulenkova@gmail.com
Comment:
You selected this USER-ID:
  "Stepan <stepannikulenkova@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? █
```

Рис. 4.4: создание ключей



Рис. 4.5: окно подтверждения

```

ap81G7D5y535J5FuD7tzoczt9YK0KASpvw+N1GwLwUC/MzJy1xAmc9nV3bjmbd3K
saJ0Nte1Mo/1vtOMAPamhF7FRZb0n33tHBLx4CZ34T+st2oMfjnd6pZQ1bNBwSH
j26p9M1a17oF2wUis3Vzh2p7Ugh02/ANF4nm1KFFA1tHhHStAivfpdHm+K4J
uWfRg6G6ck7ARk1GLm/sFB+CPmY26gtM0r+qNSRg7U1zoyeyLXKnsf4B4MnPr1
kmCeGZaTzDSQsk8dY30H+/3254cniDrpVrnd43TDEb8aKR+nh/T03AeRnuJbpc
CEXThEHPjVhEcQL5xBMA8cMoP17SngA1q+s0D2SF9v9g5X5mXv2Qetsysj1jKaR
sNaZ+Xh0fuv+hyLUFyacT0AT+Beed4XSULSScbxfz147KrT4Wk6Vvsd1CQ71XJ
T9edZ59IaH08pLmEx0+k9W5q/N7r8D7kecVVH+ERgARY3H4PgV9+GjZLhsPAM06
YXsnZBtTVSP1LXKMcOpABEBAA6Gj3VEGAIEACAWIRy9HFHrgTEYigTGKLnrrw
TyxqPuUC28w6/AIDDAAMCRCLKnrwTyxqP9BHD/4pVwmAH0FC4p1Tn/0y9nKay/b
xTW3ZNGvNEVbJ6v7bS2uyxtuvnmH49pCame3Y0V1ZG2p0GryMK+20UllwQzH8F
jgdcb3a3K1d5YwTzDpTwyj0ivQ0HhL+L37uyGfb5MhHh2DrkL3Ztu31ExmaDNSA
W9y1Be6JQd0YvH0x4jHzUG/LxPyU25QN7zqZg+meDcglVr2mIgC88/jCDM6Zi
kbtPuQZYM3o0Rwrs/D1WRK6S08IEWY/sszVALXAhhyStRm47SWS3L/zSLUAKz
EtVbYZEBkUhsqvEMOv8YlbfFgXzP6Upp14MltN8KkYUWfItXgZbbe+CNFYHoJveT
fDStwZGsy102wKvsL0aQsVaMaaUyp8eNGhgF33RoRB4xg+2MacXGWBzU3xbHAb
pMAMkAM0Ugonzlt0c1gdYoaPLexS9B13zz1FjQNS1d+qDN2aInh59GTeyhJcw/
mjMryETQDH1SGk3Ldpf8MpyatJtTb13dd/TXzfHqp1SSuQsUTSHn7bsBHBz/Cc
qrC7YPdE00tcNNDfjR8h2YmA6Mjs9ZMKF/BM12wxc/Cb+6G2DV4apTU2qA/aWce/
h8f8K681j1p56AT507ism4RHjbqkt10Y/rV71tU9z12YnaaNAcfoXReTLvZ2Z0KB
4urtinZay5yG6igBA==
-----END PGP PUBLIC KEY BLOCK-----
root@vbox:/home/ssnikulenkov# git config --global user.signingKey ^C
root@vbox:/home/ssnikulenkov# git config --global user.signingKey 8B2A7AF04F2C6A3F
root@vbox:/home/ssnikulenkov# git config --global commit.gpgsign true
root@vbox:/home/ssnikulenkov# git config --global gpg.program $(which gpg2)

```

Рис. 4.6: ключ

Выводим список ключей и копируем отпечаток приватного ключа. Скопировал сгенерированный PGP ключ в буфер обмена:

```
gpg --armor --export | xclip -sel clip
```

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

```

Available commands:
  login
  logout
  refresh
  setup-git
  status
  switch
  token

root@vbox:/home/ssnikulenkov# gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /root/.ssh/id_ed25519.pub
? Title for your SSH key: sway
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: EDD1-3844
Press Enter to open https://github.com/login/device in your browser...

```

Рис. 4.7: копирование ключа

```

GraphQL: Could not clone: Name already exists on this account (cloneTemplateRepository)
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:stepannikule
kov/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (36/36), 19.37 KiB | 483.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) re
gistered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered f
or path 'template/report'
Cloning into '/home/ssnikulenkovevbox/work/study/2024-2025/Операционные системы/os-intro/template/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 1.16 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/ssnikulenkovevbox/work/study/2024-2025/Операционные системы/os-intro/template/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)

```

Рис. 4.8: добавление ключа

Авторизуюсь через браузер создание репозитория курса на основе шаблона  
прописываю комнды для создания репозитория:

```

mkdir -p ~/work/study/2024-2025/“Операционные системы” cd ~/work/study/2024-
2025/“Операционные системы” gh repo create study_2024-2025_os-intro
-template=yamadharma/course-directory-student-template -public git clone
--recursive git@github.com:/study_2024-2025_os-intro.git os-intro

```

```

ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
config        LICENSE  package.json  README.git-flow.md  template
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ nano COURSE
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ make prepare
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am 'feat(main): make course s
tructure'
[master 577f0ef] feat(main): make course structure
405 files changed, 98413 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
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/placement_800_600_tech.jpg

```

Рис. 4.9: создание репозитория

Настройка каталога курса

Перехожу в каталог курса:

```
cd ~/work/study/2024-2025/“Операционные системы”/os-intro
```

Удаляю лишние файлы:

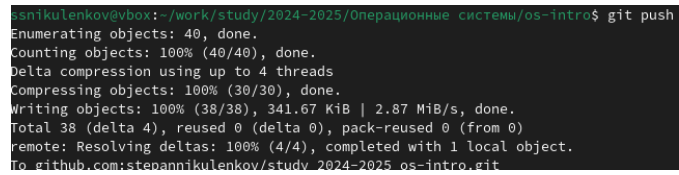
```
rm package.json
```

Создаю необходимые каталоги:

```
echo os-intro > COURSE make
```

Отправляю файлы на сервер:

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

A terminal window showing the output of a 'git push' command. The text is as follows:

```
ssnikulenkovevbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Enumerating objects: 40, done.
Counting objects: 100% (40/40), done.
Delta compression using up to 4 threads
Compressing objects: 100% (30/30), done.
Writing objects: 100% (38/38), 341.67 KiB | 2.87 MiB/s, done.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:stenannikulenkov/study_2024-2025_os-intro.git
```

Рис. 4.10: отправка на сервер

## 5 Выводы

Я изучил идеологию и применение средств контроля версий. Освоил умения по работе с git.