

Отчёт по лабораторной работе №2

Дисциплина: Архитектура Компьютеров и Операционные Системы

Мутале Чали

Содержание

1	Цель работы	1
2	Задание.....	1
3	Выполнение лабораторной работы	2
3.1	Создание базовой конфигурации для работы с git.....	2
3.2	Создание ключ ssh	2
3.3	Создание ключ gpg	3
3.4	Создание локального каталога для выполнения заданий.	6
4	Выводы.....	8
5	Ответы на контрольные вопросы	8
6	Список литературы.....	10

1 Цель работы

Изучение идеологии, применение средств контроля версий и освоение 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 Выполнение лабораторной работы

3.1 Создание базовой конфигурации для работы с git.

Устанавливаю git используя “dnf install git”:

```
cmutale@vbox:~$ sudo -i
[sudo] password for cmutale:
root@vbox:~# dnf install git
Updating and loading repositories:
Repositories loaded.
Package "git-2.48.1-1.fc41.x86_64" is already installed.

Nothing to do.
root@vbox:~#
```

Рис. 1: Установление git

С помощью dnf install gh, устанавливаю gh:

```
root@vbox:~# dnf install gh
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Installing:
gh x86_64 2.65.0-1.fc41 updates 42.6 MiB

Transaction Summary:
Installing: 1 package

Total size of inbound packages is 10 MiB. Need to download 10 MiB.
After this operation, 43 MiB extra will be used (install 43 MiB, remove 0 B).
Is this ok [y/N]: y
[1/1] gh-0:2.65.0-1.fc41.x86_64 100% | 1.3 MiB/s | 10.3 MiB | 00m08s
-----
[1/1] Total 100% | 1.1 MiB/s | 10.3 MiB | 00m09s
Running transaction
[1/3] Verify package files 100% | 8.0 B/s | 1.0 B | 00m00s
[2/3] Prepare transaction 0% | 62.0 B/s | 0.0 B | 00m00s
[0] 0:sudo* "vbox" 22:16 25-Feb-25
```

Рис. 2: Установление gh

В качестве имени и email владельца репозитории задаю свои имя и email и настраиваю utf-8:

```
cmutale@vbox:~$ git config --global user.name "cmutale-skept"
cmutale@vbox:~$ git config --global user.email "mutalechali371@gmail.com"
cmutale@vbox:~$ git config --global core.quotepath false
cmutale@vbox:~$
```

Рис. 3: имя и email владельца

Задаю имя начальной ветки и параметры autocrlf и safecrlf:

```
cmutale@vbox:~$ git config --global init.defaultBranch master
cmutale@vbox:~$ git config --global core.autocrlf input
cmutale@vbox:~$ git config --global core.safecrlf warn
cmutale@vbox:~$
```

Рис. 4: имя начальной ветки и параметры

3.2 Создание ключей ssh

Создаю ключи ssh по алгоритму rsa с размером 4096 бит:

```

root@vbox:~# 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 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:bZuR0on/EVAyNVaRmqWa9it4LxzfJY+qLwxFv9/aDn8 root@vbox
The key's randomart image is:
+---[RSA 4096]-----+
|      o.=.oo |
|      .=.o |
|      ... = |
|      +.= |
|      S.Bo.. |
|      .+=+... |
|      ***.o.* |
|      . B.o.++E |
|      ..B*.o.o= |
+-----[SHA256]-----+
root@vbox:~#

```

Рис. 5: Создание ключ ssh

3.3 Создание ключ gpg

Генерирую ключ gpg --full-generate-key:

```

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.

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

```

Рис. 6: Создание ключ gpg

Из предложенных опций выбираю тип RSA and RSA; размер 4096; срок действия 0:

```

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) y
invalid value
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

```

Рис. 7: Настройки ключ gpg

GPG запросил личную информацию, которая сохранится в ключе Имя и адрес электронной почты:

```

GnuPG needs to construct a user ID to identify your key.

Real name: cmutale-skept
Email address: mutalechali371@gmail.com
Comment:
You selected this USER-ID:
"cmutale-skept <mutalechali371@gmail.com>"

```

Рис. 8: личная информация

У меня уже есть аккаунт на github, поэтому я вхожу в систему:

The screenshot shows the GitHub profile of user **cmutale-skept**. The profile includes a custom avatar, a bio, and a button to **Edit profile**. Under **Popular repositories**, the repository **study_2024-2025_arch-pc** is listed as public. The **Contributions** section shows a calendar for 2024 with 3 contributions in the last year. The **Contribution activity** section shows no activity for February 2025 and January 2025. A link to the **GitHub profile guide** is provided at the bottom.

Рис. 9: аккаунт на git

Вывожу список ключей:

```

cmutale@vbox:~$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/21CF442F25E77A61 2025-02-25 [SC]
      5864F7977D690B9520074AF621CF442F25E77A61
uid           [ultimate] Mutale Chali <mutalechali371@gmail.com>
ssb   rsa4096/911E3457FC9BF965 2025-02-25 [E]

```

Рис. 10: список ключей

Устанавливаю xclip:

```

root@vbox:~# dnf install xclip
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Installing:
xclip x86_64 0.13-22.git11cba61.f fedora 62.4 KiB

Transaction Summary:
Installing: 1 package

Total size of inbound packages is 37 KiB. Need to download 37 KiB.
After this operation, 62 KiB extra will be used (install 62 KiB, remove 0 B).
Is this ok [y/N]: y
[1/1] xclip-0:0.13-22.git11cba61.f 100% | 80.3 KiB/s | 36.5 KiB | 00m00s
-----
[1/1] Total 100% | 20.1 KiB/s | 36.5 KiB | 00m02s
Running transaction
[1/3] Verify package files 100% | 38.0 B/s | 1.0 B | 00m00s
[2/3] Prepare transaction 100% | 0.0 B/s | 1.0 B | 00m02s
[3/3] Installing xclip-0:0.13-22.git11cba61.f 100% | 21.0 KiB/s | 64.3 KiB | 00m03s
Complete!
root@vbox:~#

```

Рис. 11: Установление xclip

Скопирую сгенерированный gpg ключ в буфер обмена:

```

cmutale@vbox:~$ gpg --armor --export 21CF442F25E77A61 | xclip -sel clip


```

Рис. 12: Копирование ключ gpg

Далее перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ:

GPG keys
New GPG key

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



new key

Email address: mutalechali371@gmail.com

Key ID: 21CF442F25E77A61

Subkeys: 911E3457FC9BF965

Added on Feb 25, 2025



Delete

Рис. 13: Добавлен ключ gpg

Используя введённый email, указываю Git применять его при подписи коммитов:

```

cmutale@vbox:~$ git config --global user.signingkey 21CF442F25E77A61
cmutale@vbox:~$ git config --global commit.gpgsign true
cmutale@vbox:~$ git config --global gpg.program $(which gpg2)
cmutale@vbox:~$

```

Рис. 14: указиваю Git

Начинаю авторизацию в gh используя gh auth login:

```

cmutale@vbox:~$ 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: 710D-4BBA
Press Enter to open https://github.com/login/device in your browser...

```

Рис. 15: авторизацию в gh

Завершаю авторизацию на броузер:

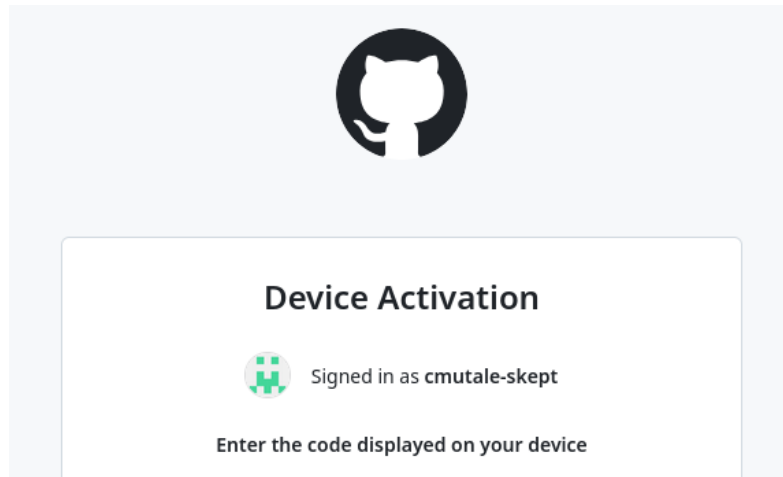


Рис. 16: Авторизоваться через броузер.

```

✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/cmutale/.ssh/id_rsa.pub
✓ Logged in as cmutale-skept
! You were already logged in to this account

```

Рис. 17: Завершена авторизация

3.4 Создание локального каталога для выполнения заданий.

Создаю каталог “mkdir -p ~/work/study/2024-2025/”Операционные системы”:

```

cmutale@vbox:~/work/study/2022-2023/Операционные системы/os-intro$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
cmutale@vbox:~/work/study/2022-2023/Операционные системы/os-intro$ cd ~/work/study/2024-2025/"Операционные системы"

```

Рис. 18: Создание каталог

Перехожу в созданный каталог:

```
cmutale@vbox:~/work/study/2024-2025/Операционные системы$ gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository cmutale-skept/study_2024-2025_os-intro on GitHub
https://github.com/cmutale-skept/study_2024-2025_os-intro
cmutale@vbox:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:cmutale/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
ERROR: Repository not found.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
cmutale@vbox:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:cmutale-skept/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), 19.38 KiB | 367.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/cmutale/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 | 338.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/cmutale/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)
Receiving objects: 100% (142/142), 341.09 KiB | 1015.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fcald4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e9495707d82ef561ab185f5c748'
cmutale@vbox:~/work/study/2024-2025/Операционные системы$
```

Рис. 19: Создание каталог

Удаляю лишние файлы:

```
cmutale@vbox:~/work/study/2024-2025/Операционные системы$ cd os-intro
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 20: Удаление файла

Создаю еще необходимые каталоги:

```
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

Targets:
  list          List of courses
  prepare       Generate directories structure
  submodule     Update submules
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 21: Создани необходимых каталогов

Отправляю Файлы на сервер:

```

cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 05dc95c] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
cmutale@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 3 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 952 bytes | 476.00 KiB/s, done.
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:cmutale-skept/study_2024-2025_os-intro.git
 89b4287..05dc95c master -> master

```

Рис. 22: Отправление файлы на сервер

4 Выводы

При выполнении лабораторной работы я изучила идеологию, применение средств контроля версий и освоила умение по работе с git.

5 Ответы на контрольные вопросы

1. Системы Контроля Версий - Программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени.
2. Хранилище - в нем хранятся все документы, включая историю их изменение и прочей служебной информацией.

commit - отслеживание измененийб сохраняет разницу в изменениях.

история - Хранит все изменения в проекте и позволяет при необходимости обратиться к нужным данным.

рабочая копия- копия проекта основанная на версии из хранилища.

3. В ценрализованном VCS например AccuRev, каждый пользователь копирует себе необходимые ему файлы из репзитория, изменяет их а затем добавляет измениения обратно в хранилище. В децентрализованном VCS например Git, есть возможность добавлять и забирать изменения из любого репозитория.
4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.

6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.

Создание основного дерева репозитория: `git init`

7. 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`

создание 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`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

6 Список литературы

::: Архитектура ЭВМ :::