

Отчет по лабораторной работе №2

Операционные системы

Подхалюзина Виолетта Михайловна

Содержание

1 Цель работы

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

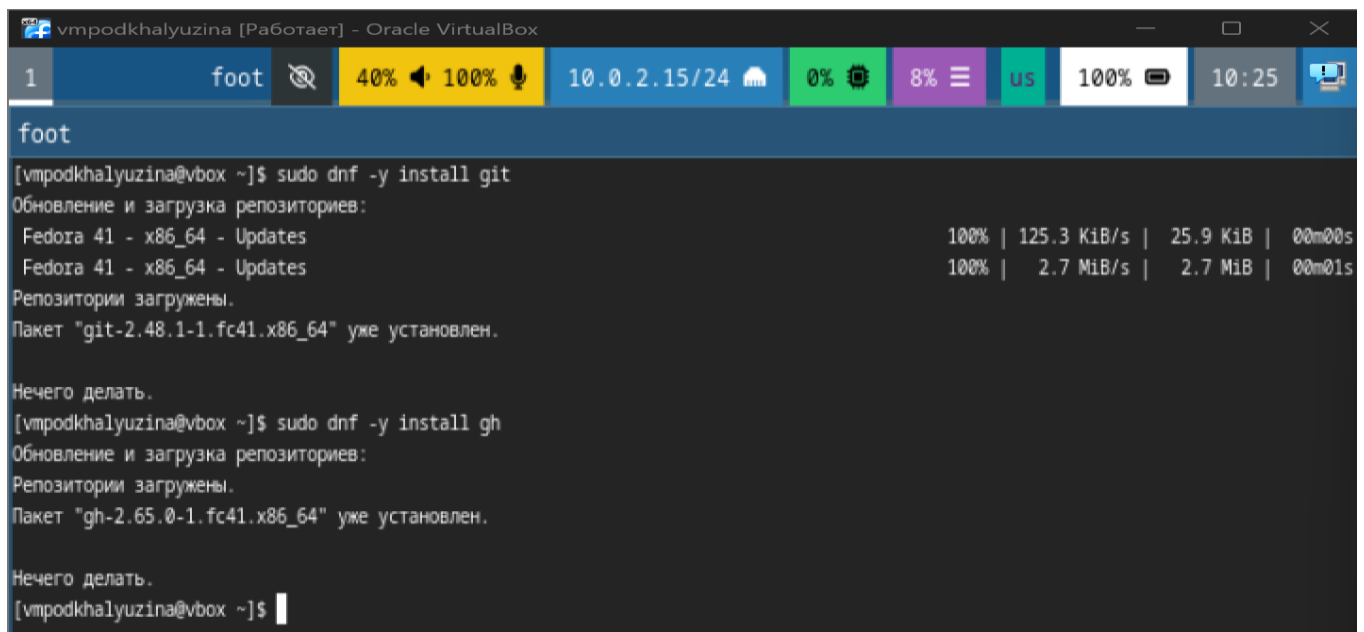
2 Задание

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.

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh` (рис. 1).



```
1 foot 40% 100% 10.0.2.15/24 0% 8% us 100% 10:25
```

```
foot
[vmpodkhalyuzina@vbox ~]$ sudo dnf -y install git
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates 100% | 125.3 KiB/s | 25.9 KiB | 00m00s
Fedora 41 - x86_64 - Updates 100% | 2.7 MiB/s | 2.7 MiB | 00m01s
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

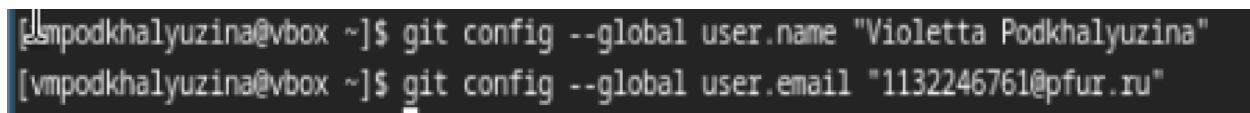
Нечего делать.
[vmpodkhalyuzina@vbox ~]$ sudo dnf -y install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "gh-2.65.0-1.fc41.x86_64" уже установлен.

Нечего делать.
[vmpodkhalyuzina@vbox ~]$
```

Figure 1: Установка git и gh

3.2 Базовая настройка git

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



```
[vmpodkhalyuzina@vbox ~]$ git config --global user.name "Violetta Podkhalyuzina"
[vmpodkhalyuzina@vbox ~]$ git config --global user.email "1132246761@pfur.ru"
```

Figure 2: Задаю имя и email владельца репозитория

Настраиваю utf-8 в выводе сообщений git для их корректного отображения (рис. 3).



```
[vmpodkhalyuzina@vbox ~]$ git config --global core.quotePath false
```

Figure 3: Настройка utf-8 в выводе сообщений git

Начальной ветке задаю имя master (рис. 4).



```
[vmpodkhalyuzina@vbox ~]$ git config --global init.defaultBranch master
```

Figure 4: Задаю имя начальной ветки

Задаю параметры autocrlf и safecrlf для корректного отображения конца строки (рис. 5).

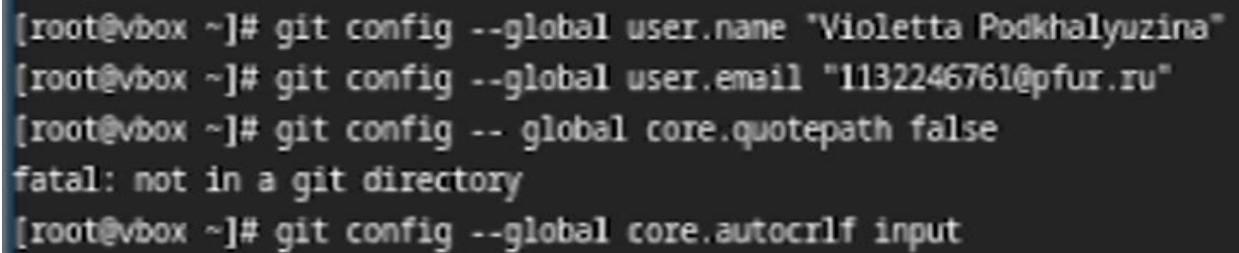


```
[vmpodkhalyuzina@vbox ~]$ git config --global core.autocrlf input
[vmpodkhalyuzina@vbox ~]$ git config --global core.safecrlf warn
```

Figure 5: Задаю параметры autocrlf и safecrlf

3.3 Создание ключа SSH

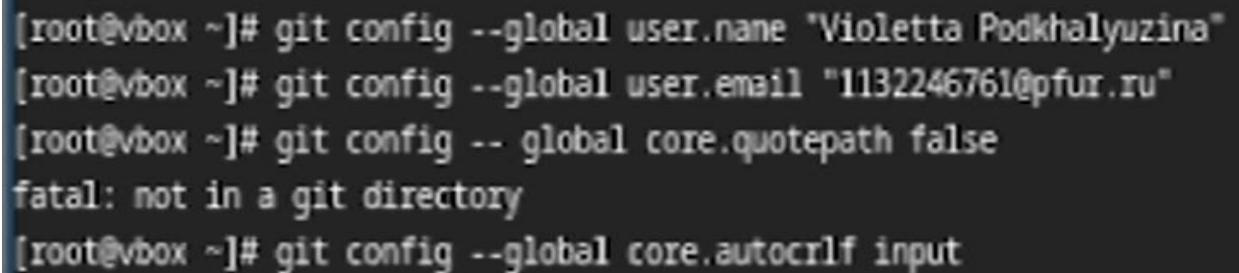
Создаю ключ ssh размером 4096 бит по алгоритму rsa (рис. 6).



```
[root@vbox ~]# git config --global user.name "Violetta Podkhalyuzina"
[root@vbox ~]# git config --global user.email "1132246761@pfur.ru"
[root@vbox ~]# git config -- global core.quotepath false
fatal: not in a git directory
[root@vbox ~]# git config --global core.autocrlf input
```

Figure 6: Генерация ssh ключа по алгоритму rsa

Создаю ключ ssh по алгоритму ed25519 (рис. 7).



```
[root@vbox ~]# git config --global user.name "Violetta Podkhalyuzina"
[root@vbox ~]# git config --global user.email "1132246761@pfur.ru"
[root@vbox ~]# git config -- global core.quotepath false
fatal: not in a git directory
[root@vbox ~]# git config --global core.autocrlf input
```

Figure 7: Генерация ssh ключа по алгоритму ed25519

3.4 Создание ключа GPG

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю максимальную длину ключа: 4096, оставляю неограниченный срок действия ключа. Далее отвечаю на вопросы программы о личной информации (рис. 8).

```

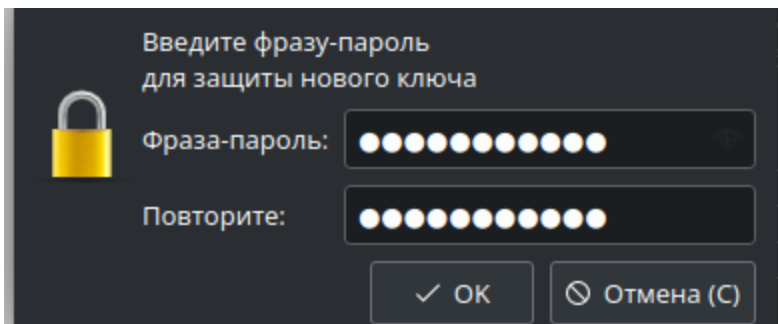
Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

```

Figure 8: Генерация ключа

Ввожу фразу-пароль для защиты нового ключа (рис. 9).



Введите фразу-пароль
для защиты нового ключа

Фраза-пароль:

Повторите:

✓ OK ⓧ Отмена (C)

Figure 9: Защита ключа GPG

3.5 Регистрация на Github

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполняла и проводила его настройку, поэтому просто вхожу в свой аккаунт (рис. 10).

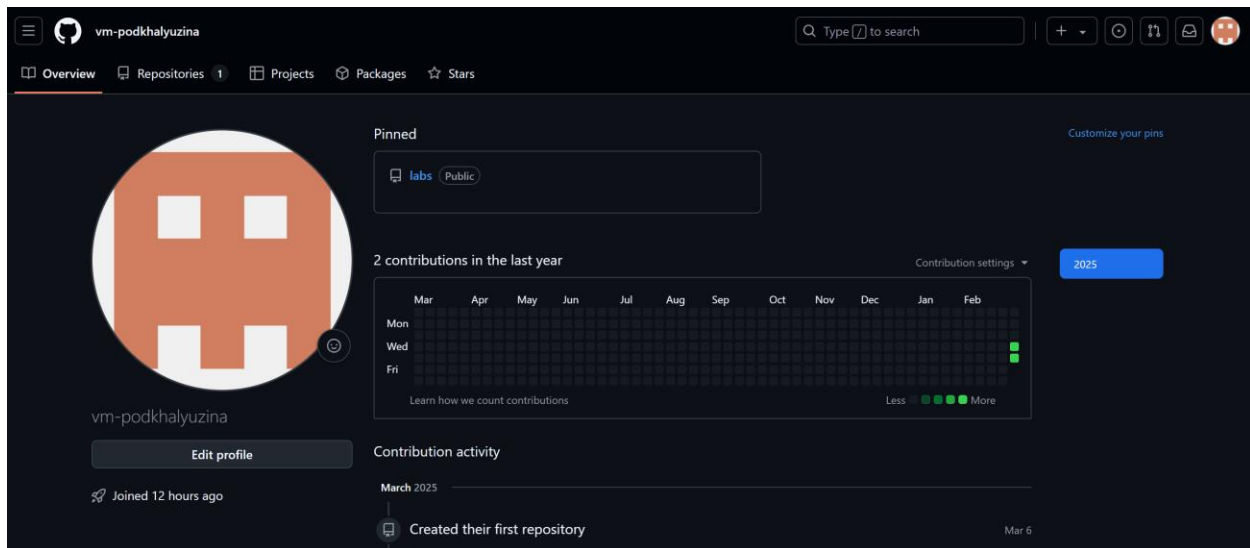


Figure 10: Аккаунт на Github

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. 11).

```
[vm-podkhalyuzina@vbox ~]$ gpg --list-secret-keys --keyid-format LONG
```

Figure 11: Вывод списка ключей

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip (рис. 12).

```
$ gpg --armor --export E2FFC767D0A4458F | xclip -sel clip
```

Figure 12: Копирование ключа в буфер обмена

Открываю настройки GitHub, ищу среди них добавление GPG ключа (рис. 13).

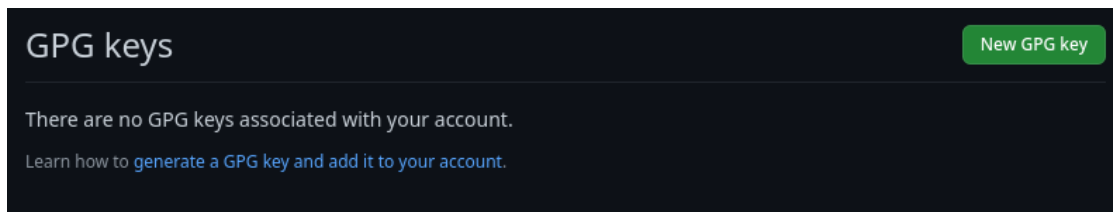


Figure 13: Настройки GitHub

Нажимаю на “New GPG key” и вставляю в поле ключ из буфера обмена (рис. 14).

```

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGfIsgIBEADvtNCPLKA14CJJ79EOhSG5RaZBpCk2spMKSoxFP1xCpHXHf5cN
KQy3XMRz5kKrCsGm2ywG951yaJpKdJS9zr07i6+ZCZXGLSUGobsKHXWJFMVYkz+W
KFydRxZ/ca/jK50Tn6mAGZX0cYzLHeR4mRqdB7LTEof89GDzrzXWMS7nBre6tcJy
w9j6uIhYzP7PChpBjWCyn4i0XHwWKAgyCsRsn1/0lxW5TPKhZqD0RyR9ioYgwF3x
TM2R7t1NC/Xeu2jZ+py2vdFMnPni2XVdm9u01xG0y59/Q6V8Ux3QV6V8DEwHrQ8p
Qf0JH1i6wPKe98JKeXre1pinX9TVYVrkiggs4PJ0XdJ1oDWTdAMdgn1AjytLIvG4
55CCsY1KOx+2fRqNdV4/eAZ1/+hfB/TFuG0i1uYRzIXH4sXPNNUo0TtXUvioY47H
uFAir03uvOEvo0p0bSZN3Nb2hRSXUZbkEr7C9RJAICrq2chjYyqQsN3ZwGBwIzAi
6DMSrRmqj2XNT44KRf41SDP+HLqNMPTYLxanrihGLPf6noyDrw7MoSsC6FGR4t
ddUSmF6JrgbGrRfswvzN5RiIrJXFy1gDhwt/Oqirfn0XtJLZfE00kEgLTndeTNHP
keeLQ8BwW21vV97H7+pMS05HRVkfso3VOQAtM016ph9X3fSJqGFxAbcd8wARAQAB
tB1Waw9sZXR0YSA8MTEzMjI0Njc2MUBwZnVyLnJlPokCUQQTAAQgA0xYhBINhw/VZ
cj6qDn0j6dPuY5e0tRmGBQJnyLICAhsDBQsJCAcCAiICBhUKCQgLAgQWAgMBAh4H
AheAAoJENPuY5e0tRmG4xcQALhkG5LWngVtU62inM6jtM4VWTZgQqt2jKwXwLDE
HilVi0PYuUdWUqQ8p+Q09ryQScQXp9hpzG20xhTvtKKPJQK10AZcT8Zq01gsU00Y
YYUf2hsbN8+oNTSRR8+FSDVibgvcExAZUqyYuL/+BEon5WDtkAxP1g/8C7NHepok
d28N117avcYscpjmv+7yYNU00srg90BrNA0U+I+eyrm2eoP038bVe00UY1sD6XXn
VQuir69pcPo3cK74TEAv+TdQDFwJHnfHdBdQjpu4CTy0zeAj2EkpTe843wodjeKi
jPLFH70CmWAmSBGZ7CsppGNvWIWmu1kc51W/fdTGSvGAPVTFPbjZ6q/iTTK604zJ
GuLDCOI3Vf/c4/B6isnugChtUpVgg0qRawFXAcYyiIMXW1IyaxRJAGxHtmtX25Za
ha+YdvjnXvVxD2Ph7140WqXiai7Tr4LzCeb3NdG52bYFY1HE3B6bykdA940n40CI
kPgrzx2D50g82RXbVN6ZqWg49zGZk7XyBwE0c62fK6P6X0R958H+H8STr21YtE+
JEF69NQo05g11xhphy1j23S/51NAPW5YmEKNUpbOrzbNRnp9wNixCmoGtXv5qGgJ
dWAGS0Y2y+qv50D5u1/7TGQ2kRZ81SVb8JFkQeQyGnm4yeA3ssb1P9ad1oeWookV
J+JwuQINBGfIsgIBEACnBDXpkhJ42Ce8PHq3/KdUm2ifM4DLcNqBq2dpDmUtMcPu
IvoV0Y54xEvSN01VaiIe4sD22Mp9mh8VL+GwAEZWK5s9+pc/kY85wda8tXwIKbWg
OxeKsDLPjrqJhFBBHIAfYJKiOZIJqKfN1z0d5FjsPu6GQsuNFLdmSFK7PiskHGUm
yodyXGqx/nMqp94mHQ1WDqeHRm4ETzcJR5XkamthRHE/J6DqFjf6eBwi9b1+YKMO
YB+K7YwM1i3+h1hIw8F0yJDVP2MvuQJns9aVYsB/GQbw170HC07nUxZGCTN1k6TI

```

Figure 14: Добавление нового PGP ключа

Я добавила ключ GPG на GitHub (рис. 15).

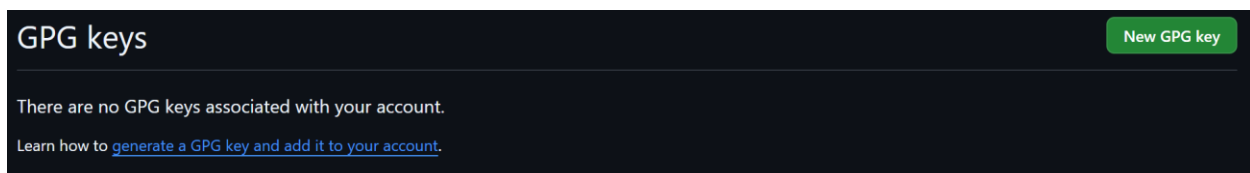


Figure 15: Добавленный ключ GPG

3.7 Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов (рис. 16).

```
[evdvorkina@evdvorkina .gnupg]$ git config --global user.signingkey E2FFC767D0A4458F
[evdvorkina@evdvorkina .gnupg]$ git config --global commit.gpgsign true
[evdvorkina@evdvorkina .gnupg]$ git config --global gpg.program $(which gpg2)
[evdvorkina@evdvorkina .gnupg]$
```

Figure 16: Настройка подписей Git

3.8 Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер (рис. 17).

```
[evdvorkina@evdvorkina .gnupg]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser
```

Figure 17: Авторизация в gh

Завершаю авторизацию на сайте (рис. 18).

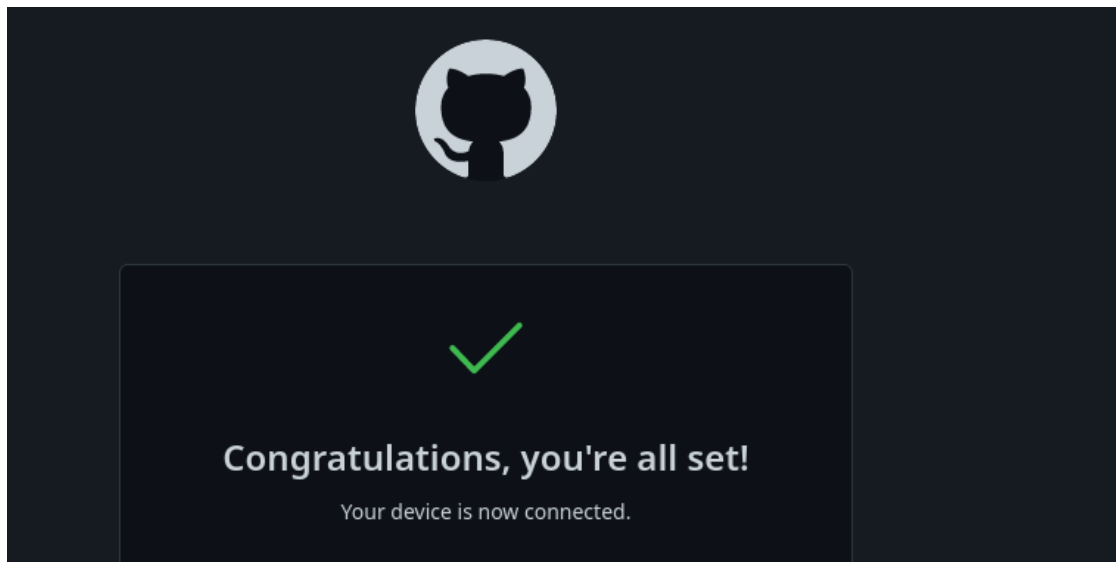


Figure 18: Завершение авторизации через браузер

Видю сообщение о завершении авторизации под именем vmpodkhalyuzina (рис. 19).


```
✓ Authentication complete.  
- gh config set -h github.com git_protocol https  
✓ Configured git protocol
```

Figure 19: Завершение авторизации

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_2022-2023_os-intro -template yamadharma/course-directory-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` выбрала протокол `https` (рис. 20).

```
Клонирование в «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 | 468.00 КиБ/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»
```

Figure 20: Создание репозитория

Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls` (рис. 21).

```
[vmpodkhalyuzina@vbox ~]$ cd os-intro  
[vmpodkhalyuzina@vbox ~]$ ls  
Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
```

Figure 21: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile` (рис. 22).

```
[vmpodkhalyuzina@vbox ~]$ rm package.json  
[vmpodkhalyuzina@vbox ~]$ echo os-intro > COURSE  
[vmpodkhalyuzina@vbox ~]$ make
```

Figure 22: Удаление файлов и создание каталогов

Добавляю все новые файлы для отправки на сервер (сохраняю добавленные изменения) с помощью команды `git add` и комментирую их с помощью `git commit` (рис. 23).

```
[vmpodkhalyuzina@vbox ~]$ git add .  
  
[vmpodkhalyuzina@vbox ~]$ git commit -am 'feat(main): make course structure'
```

Figure 23: Отправка файлов на сервер

Отправляю файлы на сервер с помощью `git push` (рис. 24).

```
[vmpodkhalyuzina@vbox ~]$ git push  
  
Перечисление объектов: 40, готово.  
Подсчет объектов: 100% (40/40), готово.  
Сжатие объектов: 100% (30/30), готово.  
Запись объектов: 100% (38/38), 343.04 КиБ | 1.67 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.
```

Figure 24: Отправка файлов на сервер

4 Выводы

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

5 Ответы на контрольные вопросы.

Системы контроля версий (VCS) — это программы, упрощающие работу с изменяющейся информацией. Они позволяют хранить несколько версий одного документа, предоставляя доступ к более ранним версиям. Такие системы удобны для коллективной работы, так как фиксируют, кто и когда вносил изменения. Основные задачи VCS — сохранение истории изменений, учет причин внесенных правок, поиск авторов изменений и организация совместной работы.

Репозиторий — это хранилище версий, включающее все документы, историю их изменений и служебные данные. Коммит фиксирует изменения, сохраняя разницу между версиями. История содержит все изменения, позволяя вернуться к любой версии. Рабочая копия — это актуальная версия проекта, загруженная из репозитория.

Централизованные VCS (например, CVS, TFS, AccuRev) используют единое хранилище проекта, из которого пользователи берут файлы, изменяют их и возвращают обратно. В децентрализованных VCS (например, Git, Bazaar) у каждого пользователя есть свой репозиторий, позволяющий обмениваться изменениями без обязательного центрального хранилища.

Сначала создается и подключается удаленный репозиторий, затем в процессе работы изменения отправляются на сервер. Перед началом работы участник проекта загружает нужную версию, вносит правки и загружает обновленную версию обратно. Все предыдущие версии сохраняются и доступны в любой момент.

Основные команды Git:

- `git init` — создание репозитория
- `git pull` — получение изменений из центрального хранилища
- `git push` — отправка локальных изменений
- `git status` — просмотр измененных файлов
- `git diff` — просмотр изменений
- `git add .` — добавление всех измененных файлов
- `git add [файл]` — добавление конкретного файла
- `git rm [файл]` — удаление файла из индекса (в локальной директории остается)
- `git commit -am "[описание]"` — сохранение изменений с комментарием
- `git commit` — сохранение изменений с вводом комментария через редактор
- `git checkout -b [ветка]` — создание новой ветки
- `git checkout [ветка]` — переключение между ветками
- `git push origin [ветка]` — отправка изменений ветки
- `git merge --no-ff [ветка]` — слияние ветки
- `git branch -d [ветка]` — удаление локальной слитой ветки
- `git branch -D [ветка]` — принудительное удаление локальной ветки
- `git push origin :[ветка]` — удаление ветки из центрального репозитория
- `git push --all` — отправка всех изменений

Ветвление — это создание параллельных участков разработки, исходящих от одной версии. Основная ветка остается стабильной, а дополнительные используются для разработки новых функций, после чего изменения сливаются.

Для исключения временных файлов из репозитория используют `.gitignore`, где указывают шаблоны игнорируемых файлов.

Список литературы

1. Лабораторная работа № 2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view.php?id=970819>