

# **Лабораторная работа №2**

**Дисциплина: Операционные системы**

Савостин Олег

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>                                       | <b>5</b>  |
| <b>2</b> | <b>Задание</b>   | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>                    | <b>8</b>  |
| 3.1      | Установка программного обеспечения . . . . .             | 8         |
| 3.2      | Базовая настройка git . . . . .                          | 8         |
| 3.3      | Создайте ключи ssh . . . . .                             | 10        |
| 3.4      | Создайте ключи pgr . . . . .                             | 10        |
| 3.5      | Настройка github . . . . .                               | 11        |
| 3.6      | Добавление PGP ключа в GitHub . . . . .                  | 12        |
| 3.7      | Настройка автоматических подписей коммитов git . . . . . | 13        |
| 3.8      | Настройка gh . . . . .                                   | 14        |
| 3.9      | Шаблон для рабочего пространства . . . . .               | 14        |
| <b>4</b> | <b>Выводы</b>  | <b>17</b> |
|          | <b>Список литературы</b>                                 | <b>18</b> |

# Список иллюстраций

|      |   |    |
|------|---|----|
| 3.1  | Установка программного обеспечения . . . . .          | 8  |
| 3.2  | Задаю имя и почту владельца . . . . .                 | 8  |
| 3.3  | utf-8 . . . . .                                       | 9  |
| 3.4  | Верификация и подписание коммитов . . . . .           | 9  |
| 3.5  | Записывание параметров . . . . .                      | 9  |
| 3.6  | Создание ключей ssh . . . . .                         | 10 |
| 3.7  | gpg –full-generate-key . . . . .                      | 11 |
| 3.8  | Аккаунт на гитхабе . . . . .                          | 12 |
| 3.9  | Копирую ключ . . . . .                                | 13 |
| 3.10 | Новый ключ gpg на аккаунте . . . . .                  | 13 |
| 3.11 | Email commit . . . . .                                | 13 |
| 3.12 | gh auth login . . . . .                               | 14 |
| 3.13 | Создание рабочего пространства . . . . .              | 14 |
| 3.14 | Удаление лишних файлов и создание каталогов . . . . . | 15 |
| 3.15 | Создание каталогов . . . . .                          | 16 |
| 3.16 | Отправка файлов на сервер . . . . .                   | 16 |

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создайте ключи ssh
4. Создайте ключи pgp
5. Настройка github
6. Добавление PGP ключа в GitHub
7. Настройка автоматических подписей коммитов git
8. Настройка gh
9. Шаблон для рабочего пространства # Теоретическое введение

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.

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

В первую очередь, устанавливаем git и gh(рис. 3.1).

```
[root@vbox ~]# dnf install git
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.46.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@vbox ~]# dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
Аrch. Версия Репозиторий Размер
gh x86_64 2.65.0-1.fc41 updates 42.6 MiB
Сводка транзакции:
Установка: 1 пакета
Общий размер входящих пакетов составляет 18 MiB. Необходимо загрузить 18 MiB.
После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] gh-2.65.0-1.fc41.x86_64 100% | 118.9 KiB/s | 10.3 MiB | 01m29s
[1/1] Total 100% | 114.0 KiB/s | 10.3 MiB | 01m33s
Выполнение транзакции
```

Рис. 3.1: Установка программного обеспечения

### 3.2 Базовая настройка git

Задаю имя и почту владельца репозитория. Так как у меня уже есть аккаунт на github'е, то я задаю данные, которые я вводил когда создавал аккаунт(рис. 3.2).

```
Завершено!
[root@vbox ~]# git config --global user.name "Oleg Savostin"
[root@vbox ~]# git config --global user.email "mr.sans.san2@gmail.com"
[root@vbox ~]# git config --global core.quotepath false
```

Рис. 3.2: Задаю имя и почту владельца

Настраиваю utf-8 в выводе сообщений гит, настраиваю верификацию и подписание коммитов git(рис. 3.3) (рис. 3.4).



```
[root@vbox ~]# git config --global user.name "osavostin"
[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: создан каталог '/root/.gnupg'
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? █
```

Рис. 3.3: utf-8

```
Ваше полное имя: Savostin Oleg
Адрес электронной почты: mr.sans.san2@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Savostin Oleg <mr.sans.san2@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/A796C4823E0BCACABE162E21DF6851E9AB59EB53.rev'.
Открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2025-03-02 [SC]
     A796C4823E0BCACABE162E21DF6851E9AB59EB53
uid          Savostin Oleg <mr.sans.san2@gmail.com>
sub  rsa4096 2025-03-02 [E]

[root@vbox ~]# █
```

Рис. 3.4: Верификация и подписание коммитов

Записываю параметры autocrlf safecrlf(рис. 3.5).

```
[root@vbox ~]# git config --global init.defaultBranch master
[root@vbox ~]# git config --global core.autocrlf input
[root@vbox ~]# git config --global core.safecrlf warn
```

Рис. 3.5: Записывание параметров

### 3.3 Создайте ключи ssh

По алгоритму rsa с ключём размером 4096 создаю ключ. Также создаю ключ по алгоритму ed25519(рис. 3.6).

```
[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:hydMoih0XrkghXqEXaKgnzMyFXSYt84BNPd6dN9ij6U root@vbox
The key's randomart image is:
+----[RSA 4096]-----+
|oo..ooo=|
| oo.o++* o|
|.. .+++.o|
|...O+.oo...|
| +*.no S+oo|
| oo+.. .+*|
|o.o . E .|
|...|
|
+-----[SHA256]-----+
[root@vbox ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
```

Рис. 3.6: Создание ключей ssh

### 3.4 Создайте ключи prg

Следуя инструкциям, указанных в лабораторной работе, создаю ключ prg prg -full-generate-key (рис. 3.7).

```
[root@vbox ~]# gpg --armor --export DF6851E9AB59EB53
-----BEGIN PGP PUBLIC KEY BLOCK-----

nQINB6fETvUBiEADv14RnJo0WV3LCUbaLyyvoFryuwtq8Zc6fzqZfo88BVPhDILRR
SCJpbj5RYz7XkbWUYJJ0INncXUW0jvW5/dZ4TXy7MpJmVT7U51owkQb0jcIPTT
K9oCyLP8JaiidLW0Dj7VH/n3jb/cVvQBv0Xp6HmJ4m3RPZUAP7TtIWZnf9nR01Jc
71PF9yEjc5IxHxS+PEF/1RxdGAzmzHuRK6xPJXLd6Lzo/wAJzV4RItItF97EgzE1
9y0w1Ewuq+zGLGktEbttnL1gdA3Ea+0TvAAobUocekPznJr1gbKsbacQj11S37uB
NM6SvipgH+bWnjIIR5XPcrj4GvLEqTbrZqQ517p0NjCNBngHhvPLlpk2vEdZw0fU
3IKGb5hJFBDY0fagy/vPuzC0UJ1fZqZ4vW56Y14J0UhfRByZtySne0x9R3ub/rzP
On70KmHXQMyriqB3MWctgHUGLx++LurFITb1st6pDkpsaUxIyNt2ZjWraOkBP009d
vYnkC1jyIp2TqYCNLdsbsT1Svu5WGUlqLgiKAeyW5IGmWZKwz7Y20Kz0RNPmftA
/6IudlogyzX2mI15V1x0v1Rs4CQAsC4hSy0o+Iv053NkCgVfmKCKDkS1KD0E7ipj
72c5Z6nA14UNCTB6ZwbKh5BmhwMtAuuoVE90zAjsSaxVFYmwb0s39Q08hwARAQAB
tCZTYXZvc3RpbjBPbGVnIDxtci5zYW5zLnNhbjJAZ21haWwY29tPokCUQQAQgA
OxYhBKEwXII+C8zKvhYuId9oUenrWetTBQJnxE71AhsDBQsJCAcCA1ICBhUKCQGL
AgQwAqMBAh4HAheAAoJEN9oUenrWetT0t8P/AqM8kCQzqcVLGxrZrcTZ1jx33v5
bbH0hW/n/r4xNypIqsFKkNbahp9jnb13ad/Re8+n1uwSMVT7Q4o1MY1M3fIfGa
UmPv6NkMny3e3TW7pE0Yw0GJrGyDQHf0H+hEr+wVxXzSmkocTZGQNe7P0Vje5j3
yf7+g1i/n1EvF/bmXcG45HEN0U+v8XhWtIKY9Dfgh38shMogIX40uu06HkoqYInt
qBE2XfV2132qyK/O1x5LU2XeJeJasfA5Qym576Gt1VNXnFukP/vcFTGKKIUjV8r9
NMWQsXDPmg681bwQd1EoJBfX76PwN4H+A3ZUbJ0UPANhFFcUfYr0VbY1YH21KK
cDqnqo9WymOG9PoEVSIvAIF5TsUFY1CPDjTIMNfC/VX7URg+Ddv7D6aDEZjJ83yu
81h8VGk0rWgZwzyIRNwxLTDq9m7gUduYcKg8Q54UteR9SKDtvEw8Sg38xJYnnoIb
```

Рис. 3.7: gpg –full-generate-key

## 3.5 Настройка github

Аккаунт заранее существовал с первого семестра. (рис. 3.8).

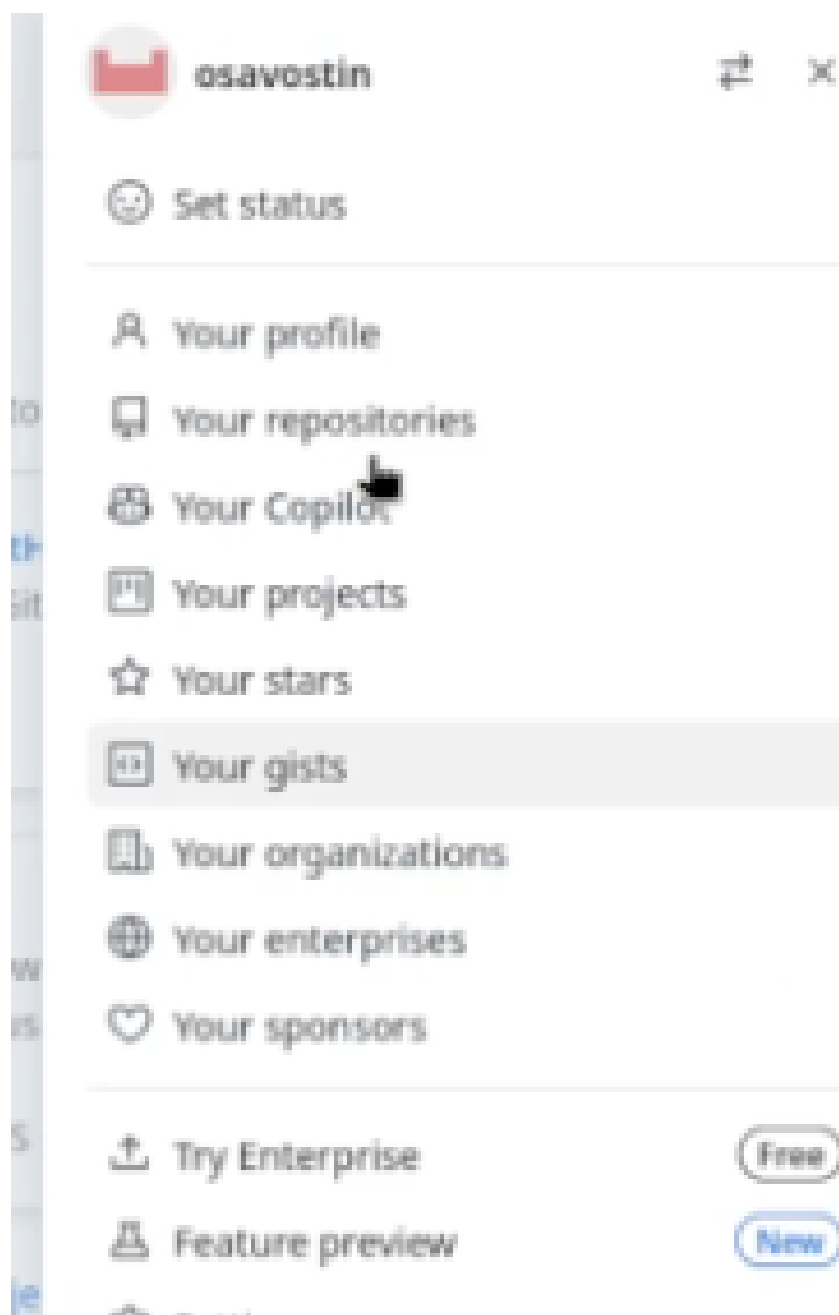


Рис. 3.8: Аккаунт на гитхабе

### 3.6 Добавление PGP ключа в GitHub

Вывожу список ключей и копирую отпечаток приватного ключа командой `gpg --list-secret-keys --keyid-format LONG` (рис. 3.9).

```
[root@vbox ~]# gpg --armor --export DF6851E9AB59EB53 | xclip -sel clip
```

Рис. 3.9: Копирую ключ

Копирую отпечаток ключа и вставляю его в настройки GitHub (рис. 3.10).

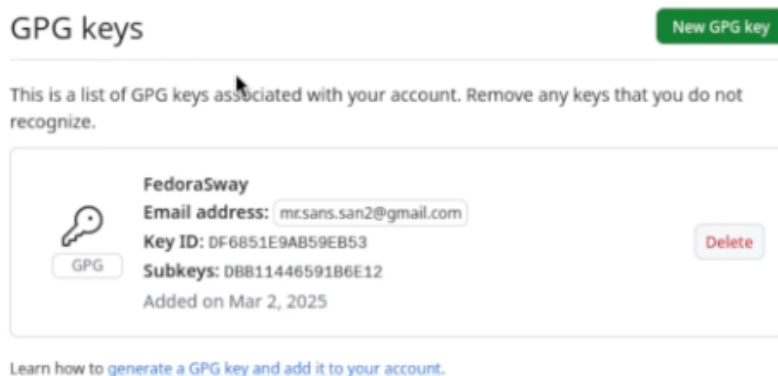


Рис. 3.10: Новый ключ gpg на аккаунте

## 3.7 Настройка автоматических подписей коммитов git

Используя введённый email, указываю Гит применять его при подписи коммитов (рис. 3.11).

```
[root@vbox ~]# git config --global user.signingkey DF6851E9AB59EB53
[root@vbox ~]# git config --global commit.gpg.sign true
[root@vbox ~]# git config -- global.gpg.program $(which gpg2)
fatal: not in a git directory
[root@vbox ~]# git config --global gpg.program $(which gpg2)
[root@vbox ~]# 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: Key
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token:
```

Рис. 3.11: Email commit

## 3.8 Настройка gh

Для пользования `gh` я захожу в свой аккаунт через терминал (рис. 3.12).

```
[root@vbox ~]# git config --global user.signingkey DF6851E9AB59EB53
[root@vbox ~]# git config --global commit.gpg.sign true
[root@vbox ~]# git config --global gpg.program $(which gpg2)
fatal: not in a git directory
[root@vbox ~]# git config --global gpg.program $(which gpg2)
[root@vbox ~]# 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: Key
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: 
```

Рис. 3.12: `gh auth login`

## 3.9 Шаблон для рабочего пространства

Теперь необходимо создать шаблон рабочего пространства. Пишу в терминал следующие команды: `mkdir -p ~/work/study/2022-2023/“Операционные системы”`  
`cd ~/work/study/2022-2023/“Операционные системы”`  
`gh repo create study_2022-2023_os-intro --template=yamadharm/course-directory-student-template --public`  
`git clone --recursive git@github.com:/study_2022-2023_os-intro.git os-intro` (рис. 3.13).

```
root@vbox:~# mkdir -p ~/work/study/2024-2025/“Операционные системы”
root@vbox:~# cd ~/work/study/2024-2025/“Операционные системы”
root@vbox:~/work/study/2024-2025/“Операционные системы”# gh repo create study_2024-2025_os-intro --template=yamadharm/course-directory-student-template --public
Created repository osavostin/study_2024-2025_os-intro on GitHub
https://github.com/osavostin/study_2024-2025_os-intro
root@vbox:~/work/study/2024-2025/“Операционные системы”# git clone --recursive git@github.com:osavostin/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
ssh: connect to host github.com port 22: Connection refused
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
root@vbox:~/work/study/2024-2025/“Операционные системы”# git clone --recursive mr.sans.san2@gmail.com:osavostin/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
```

Рис. 3.13: Создание рабочего пространства

Перехожу в каталог курса и удаляю лишние файлы `.json` и затем создаю нужные каталоги (рис. 3.14) (рис. 3.15).

```

root@vbox:~# cd ~/work/study/2024-2025/"Операционные системы"
root@vbox:~/work/study/2024-2025/Операционные системы# cd os-intro
root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# rm package.json
rm: удалить обычный файл 'package.json'? y
root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# echo os-intro > COURSE
root@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

root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# make list
net-admin      Администрирование локальных сетей
net-os-admin   Администрирование сетевых подсистем
arch-pc        Архитектура ЭВМ
sciprogram-intro Введение в научное программирование
netcybersec    Защита сетей и кибербезопасность
simmod         Имитационное моделирование
infosec        Информационная безопасность
computer-practice Компьютерный практикум по статистическому анализу данных
mathsec        Математические основы защиты информации и информационной безопасности
mathmod        Математическое моделирование
simulation-networks Моделирование сетей передачи данных
sciprogram     Научное программирование
os-intro       Операционные системы
os2            Основы администрирования операционных систем
infosec-intro  Основы информационной безопасности
root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# make prepare
root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# make submodule

```

Рис. 3.14: Удаление лишних файлов и создание каталогов

```

root@vbox:~# 100755 project-personal/stage4/report/pandoc/filters/pandoc_secnos.py
[sudo] napde 100755 project-personal/stage4/report/pandoc/filters/pandoc_tablenos.py
root@vbox:~# 100644 project-personal/stage4/report/pandoc/filters/pandocxnos/__init__.py
root@vbox:~# 100644 project-personal/stage4/report/pandoc/filters/pandocxnos/core.py
root@vbox:~# 100644 project-personal/stage4/report/pandoc/filters/pandocxnos/main.py
rm: удалите 100644 project-personal/stage4/report/pandoc/filters/pandocxnos/pandocattributes.py
root@vbox:~# 100644 project-personal/stage4/report/report.md
root@vbox:~# 100644 project-personal/stage5/presentation/.projectile
Usage: 100644 project-personal/stage5/presentation/.texlabroot
make <100644 project-personal/stage5/presentation/Makefile
100644 project-personal/stage5/presentation/image/kulyabov.jpg
Targets: 100644 project-personal/stage5/presentation/presentation.md
list 100644 project-personal/stage5/report/Makefile
prepare 100644 project-personal/stage5/report/bib/cite.bib
submodu 100644 project-personal/stage5/report/image/placeimg_800_600_tech.jpg
100644 project-personal/stage5/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
root@vbox:~# 100755 project-personal/stage5/report/pandoc/filters/pandoc_eqnos.py
100755 project-personal/stage5/report/pandoc/filters/pandoc_fignos.py
netde 100755 project-personal/stage5/report/pandoc/filters/pandoc_secnos.py
100755 project-personal/stage5/report/pandoc/filters/pandoc_tablenos.py
scipde 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/__init__.py
node 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/core.py
100644 project-personal/stage5/report/pandoc/filters/pandocxnos/main.py
100644 project-personal/stage5/report/pandoc/filters/pandocxnos/pandocattributes.py
compute 100644 project-personal/stage5/report/report.md
100644 project-personal/stage6/presentation/.projectile
100644 project-personal/stage6/presentation/.texlabroot
simulation 100644 project-personal/stage6/presentation/Makefile
100644 project-personal/stage6/presentation/image/kulyabov.jpg
100644 project-personal/stage6/presentation/presentation.md
100644 project-personal/stage6/report/Makefile
info 100644 project-personal/stage6/report/bib/cite.bib
root@vbox:~# 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg
root@vbox:~# 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
100755 project-personal/stage6/report/pandoc/filters/pandoc_eqnos.py
git submo 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
Entering 100755 project-personal/stage6/report/pandoc/filters/pandoc_secnos.py
fatal: 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
Указатель 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
Entering 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
fatal: 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
100644 project-personal/stage6/report/report.md

```

Рис. 3.15: Создание каталогов

Успешно отправляю файлы на сервер (рис. 3.16).

```

root@vbox:~/work/study/2024-2025/Операционные системы/os-intro# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 341.66 КиБ | 555.00 КиБ/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object

```

Рис. 3.16: Отправка файлов на сервер



## 4 Выводы

В ходе проделанной работы я изучил идеологию и применение средств контроля версий и освоил умения по работе с git.

# **Список литературы**

Лабораторная работа №2