

# **Отчёт по лабораторной работе №2**

**Операционные системы**

Терещенкова М.В.

# 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>
4.1	Установка программного обеспечения . . . . .	8
4.2	Базовая настройка git . . . . .	8
4.3	Создание ключа SSH . . . . .	9
4.4	Создание ключей pgr . . . . .	10
4.5	Настройка github. . . . .	11
4.6	Добавление PGP ключа в GitHub . . . . .	12
4.7	Настройка автоматических подписей коммитов git . . . . .	13
4.8	Настройка gh . . . . .	13
4.9	Создание репозитория курса на основе шаблона . . . . .	15
4.10	Настройка каталога курса . . . . .	16
<b>5</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1	Установка git и gh . . . . .	8
4.2	Задаю имя и email владельца репозитория; настраиваю utf-8 в выводе сообщений git; задаю имя начальной ветки и параметры autocrlf и safecrlf . . . . .	9
4.3	Создание ключа ssh по алгоритму rsa . . . . .	9
4.4	Создание ключа ssh по алгоритму ed25519 . . . . .	10
4.5	Создание ключа pgr . . . . .	11
4.6	Мой профиль в github . . . . .	12
4.7	Вывод списка ключей . . . . .	12
4.8	Команда копирования . . . . .	12
4.9	New GPG key . . . . .	13
4.10	Настройка автоматических подписей коммитов git . . . . .	13
4.11	Авторизация gh . . . . .	13
4.12	one-time code . . . . .	14
4.13	Успешная авторизация . . . . .	14
4.14	Успешная авторизация.2 . . . . .	15
4.15	Создание репозитория . . . . .	15
4.16	Создание репозитория.2 . . . . .	15
4.17	Перемещение между директориями . . . . .	16
4.18	Удаление файлов и создание каталогов . . . . .	16
4.19	Отправка файлов на сервер . . . . .	16
4.20	Отправка файлов на сервер.2 . . . . .	16

# 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 GPG
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.

### **Примеры использования git**

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

## 4 Выполнение лабораторной работы

### 4.1 Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: **dnf install git** и **dnf install gh**

```
root@teleshnikovmagnit:~# dnf install git
Для выполнения запрошенной операции требуется привилегия суперпользователя. Пожалуйста, войдите в систему как пользователь с повышенными правами или используйте опции "--assume-yes" или "--no-
installonly", чтобы выполнить команду без изменения состояния системы.
root@teleshnikovmagnit:~# sudo -i
[sudo] пароль для teleshnikovmagnit:
root@teleshnikovmagnit:~# dnf install git
Обновление и загрузка репозитория:
Репозиторий загружен.
Пакет "git-2.40.1-1.fc41.x86_64" уже установлен.
Ничего делать.
root@teleshnikovmagnit:~# dnf install gh
Обновление и загрузка репозитория:
Репозиторий загружен.
Пакет
Установка:
Аrch. Версия Репозиторий Размер
Установка: x86_64 2.65.0-1.fc41 updates 42.6 MiB
Сводка транзакции:
Установка: 1 пакета
Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.
После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB, удаление 0 B).
Is this ok [Y/n]: Y
```

Рис. 4.1: Установка git и gh

### 4.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту. Настраиваю utf-8 в выводе сообщений git для их корректного отображения. Настраиваю utf-8 в выводе сообщений git для их корректного отображения. Настраиваю utf-8 в выводе сообщений git для их корректного отображения. Начальной ветке задаю имя master. Задаю параметры autocrlf и safecrlf для корректного отображения конца строки.



```

root@tereshchenkovamargo:~# git config --global user.name "Rita Tereshchenkova"
root@tereshchenkovamargo:~# git config --global user.email "1132246723@yur.ru"
root@tereshchenkovamargo:~# git config --global core.quotepath false
root@tereshchenkovamargo:~# git config --global init.defaultBranch master
root@tereshchenkovamargo:~# git
git-s-tool      git-querymd5-64  git-latestdiff  git-shell        git-upload-pack
git             git              git-receive-pack  git-upload-archive
root@tereshchenkovamargo:~# git config --global core.autocrlf input
root@tereshchenkovamargo:~# git config --global core.safecrlf warn

```

Рис. 4.2: Задаю имя и email владельца репозитория; настраиваю utf-8 в выводе сообщений git; задаю имя начальной ветки и параметры autocrlf и safecrlf

## 4.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa.

```

root@tereshchenkovamargo:~# 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:lftbqlyFDnpx599/jJTfEUj04rCdxCltuKzrr2lfbNE root@tereshchenkovamargo.net
The key's randomart image is:
+---[RSA 4096]---+
|          oo  |
|         oo.o |
|        ++o=. |
|       .+0o+. |
|      S=.o=oE. |
|     o*.++. |
|    .o +=o+o |
|   =..o+..* |
|  o*+o  * |
+-----[SHA256]-----+

```

Рис. 4.3: Создание ключа ssh по алгоритму rsa

Создаю ключ ssh по алгоритму ed25519

```
[root@tereshchenkovamargo:~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase for "/root/.ssh/id_ed25519" (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:
SHA256:6c4H/xFQ2GeQV+S0dJuhK8C0lf2ZVE8C56K7qopN49Q root@tereshchenkovamargo.net
The key's randomart image is:
+--[ED25519 256]--+
|      .  +*+o=0|
|      o  +oo+**B|
|      +.  .+=+=|
|      ..O  ..+ |
|      S  .... |
|      .  .  .  .|
|      + E  .O.  .|
|      *  .  o  o.  .|
|      .  +...+o...|
+-----[SHA256]-----+
root@tereshchenkovamargo:~#
```

Рис. 4.4: Создание ключа ssh по алгоритму ed25519

## 4.4 Создание ключей pgr

Генерируем ключ. Из предложенных опций выбираю: тип RSA and RSA; размер 4096; выбираю срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). Также ввожу своё имя и корпоративную почту.

```
root@tereshchenkovamargo:~# 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.

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.

Ваше полное имя: TereshchenkovaRita
Адрес электронной почты: 
```

Рис. 4.5: Создание ключа pgp

## 4.5 Настройка github.

У меня уже настроен github.

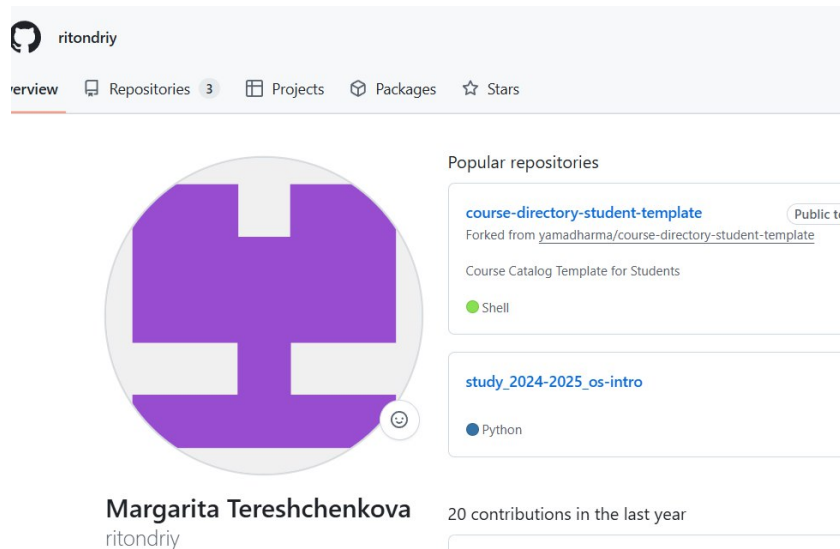


Рис. 4.6: Мой профиль в github

## 4.6 Добавление PGP ключа в GitHub

Выводим список ключей и копируем отпечаток приватного ключа с помощью команды: **gpg --list-secret-keys --keyid-format LONG**

```
root@tereshchenkovamargo:~# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 2 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 2u
[keyboard]
-----
sec   rsa4096/831708CE8986E5B9 2025-03-05 [SC]
      3C67D5A97DDBD25D616127DB831708CE8986E5B9
uid   [ абсолютно ] TereshchenkovaRita <1132246723@pfur.ru>
ssb   rsa4096/DF2FDA6701EEBC5A 2025-03-05 [E]

sec   rsa4096/C154FBFC532754BE 2025-03-05 [SC]
      77C5B195FEC56E4E8F8C2D2CC154FBFC532754BE
uid   [ абсолютно ] TereshchenkovaRita <1132246723@pfur.ru>
ssb   rsa4096/3FE41D552B521EB7 2025-03-05 [E]
```

Рис. 4.7: Вывод списка ключей

Скопировала свой сгенерированный PGP ключ в буфер обмена: **gpg --armor --export C154FBFC532754BE |xclip -sel clip**

```
gpg: filter_flush failed on close: Обрыв канала
root@tereshchenkovamargo:~# gpg --armor --export C154FBFC532754BE |xclip -sel clip
root@tereshchenkovamargo:~# gpg --armor --export C154FBFC532754BE |xclip -sel clip
```

Рис. 4.8: Команда копирования

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

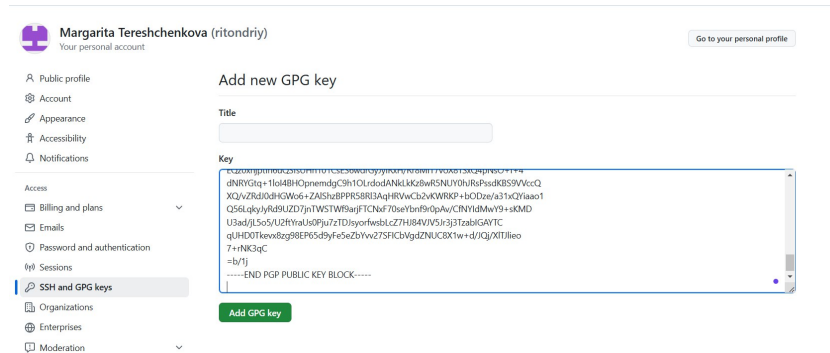


Рис. 4.9: New GPG key

## 4.7 Настройка автоматических подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов.

```
root@tereshchenkovamargo:~# git config --global user.signingkey C154FBFC532754BE
root@tereshchenkovamargo:~# git config --global commit.gpgsign true
root@tereshchenkovamargo:~# git config --global gpg.program $(which gpg2)
```

Рис. 4.10: Настройка автоматических подписей коммитов git

## 4.8 Настройка gh

Авторизовываюсь с помощью команды: **gh auth login** и отвечаю на вопросы утилиты.

```
root@tereshchenkovamargo:~# 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: 889F-183C
Press Enter to open https://github.com/login/device in your browser...
restorcom: SELinux: Could not get canonical path for root/.mozilla/firefox/*gmp-widevinecdm/*restorcom: No such file or directory.
Running Firefox as root in a regular user's session is not supported. (EXAUTHORITY is /run/lightdm/tereshchenkovamargo/authority which is owned by tereshchenkovamargo.)
```

Рис. 4.11: Авторизация gh

Ввожу код.

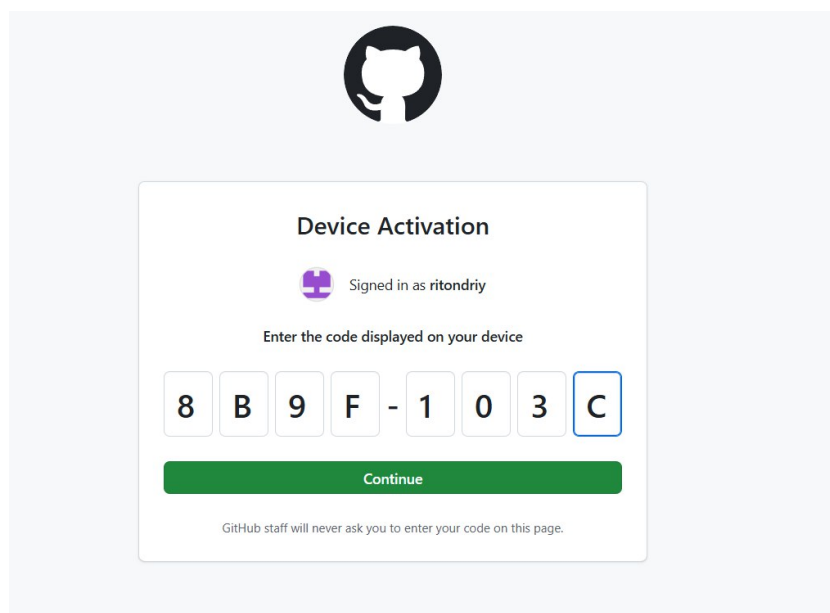


Рис. 4.12: one-time code

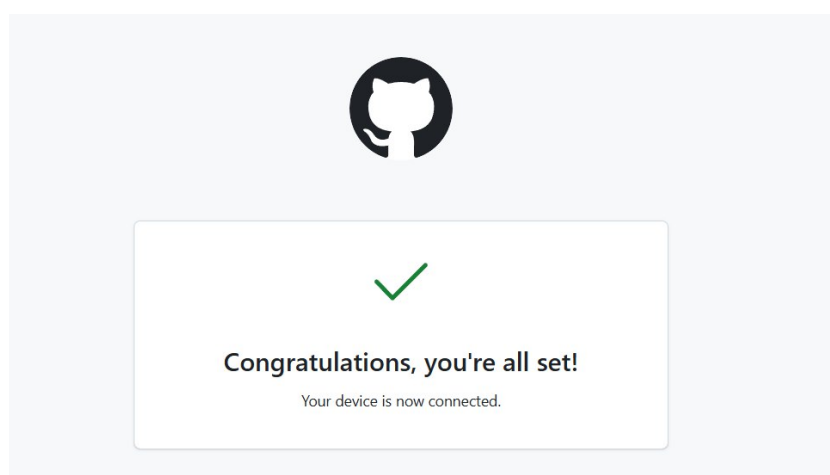


Рис. 4.13: Успешная авторизация

```

! First copy your one-time code: 8B9F-103C
Press Enter to open https://github.com/login/device in
restorecon: SELinux: Could not get canonical path for /
Running Firefox as root in a regular user's session is

✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as ritondriy
root@tereshchenkovamargo:~#

```

Рис. 4.14: Успешная авторизация.2

## 4.9 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_2024-2025_os-intro-template yamadharma/course-directory-student-template --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`.

```

root@tereshchenkovamargo:~# mkdir -p /work/study/2024-2025/Операционные системы
root@tereshchenkovamargo:~# cd /work/study/2024-2025/Операционные системы
root@tereshchenkovamargo:/work/study/2024-2025/Операционные системы# gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
Created repository ritondriy/study_2024-2025_os-intro on GitHub
https://github.com/ritondriy/study_2024-2025_os-intro
root@tereshchenkovamargo:/work/study/2024-2025/Операционные системы# git clone --recursive https://github.com/ritondriy/study_2024-2025_os-intro.git os-intro

```

Рис. 4.15: Создание репозитория

```

https://github.com/ritondriy/study_2024-2025_os-intro.git
root@tereshchenkovamargo:/work/study/2024-2025/Операционные системы# git clone --recursive https://github.com/ritondriy/study_2024-2025_os-intro.git os-intro
Клонирование в «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)
Получение объектов: 100% (36/36), 19.38 KiB | 2.42 MiB/c, готово.
Определение изменений: 100% (1/1), готово.
Настройка шаблона presentation: (https://github.com/yamadharma/academic-presentation-markdown-template.git) аперскриптован на нуть «template/presentation»
Настройка шаблона report: (https://github.com/yamadharma/academic-laboratory-report-template.git) аперскриптован на нуть «template/report»
Клонирование в «/root/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% (107/107), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 182.17 KiB | 141.00 KiB/c, готово.
Определение изменений: 100% (52/52), готово.
Клонирование в «/root/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 68), reused 123 (delta 39), pack-reused 0 (from 0)
Получение объектов: 100% (142/142), 341.09 KiB | 981.00 KiB/c, готово.
Определение изменений: 100% (68/68), готово.
Submodule path 'template/presentation': checked out 'c902712b4b20431a05886c9c72a02d2fca1d4a6'
Submodule path 'template/report': checked out 'c26a22effe7b3a8495787d3a7561ab185f5c748'

```

Рис. 4.16: Создание репозитория.2

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

```
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы# cd os-intro
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  package.json  README.en.md  README.git-flow.md  README.md  template
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro#
```

Рис. 4.17: Перемещение между директориями

## 4.10 Настройка каталога курса

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile` и проверяю корректность выполнения с помощью утилиты `ls`.

```
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# rm package.json
rm: удалить обычный файл 'package.json'? y
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# echo os-intro > COURSE
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# make prepare
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# ls
CHANGELOG.md  config  COURSE  labs  LICENSE  Makefile  prepare  presentation  project-personal  README.en.md  README.git-flow.md  README.md  template
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro#
```

Рис. 4.18: 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`.

```
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# git add .
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# git commit -am 'feat(main): make course structure'
[master c04af1f] 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/teclabroot
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/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
```

Рис. 4.19: Отправка файлов на сервер

Отправляю файлы на сервер с помощью `git push`

```
root@tereshchenkovamargo:~/work/study/2024-2025/Операционные системы/os-intro# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.32 КиБ | 4.08 МБ/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.
To https://github.com/ritondriy/study_2024-2025_os-intro.git
917a06d..c04af1f  master -> master
```

Рис. 4.20: Отправка файлов на сервер.2



## **5 Выводы**

Изучила идеологию и применение средств контроля версий; освоила умения по работе с git.

## **Список литературы**