

# **Отчет по лабораторной работе №2**

**Операционные системы**

Сокирка Анна Константиновна

# 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>7</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>
<b>5</b>	<b>Ответы на контрольные вопросы.</b>	<b>18</b>
<b>6</b>	<b>Список литературы</b>	<b>21</b>

# Список иллюстраций

3.1	Установка git и gh . . . . .	7
3.2	Задаю имя и email владельца репозитория . . . . .	7
3.3	Настройка utf-8 в выводе сообщений git . . . . .	8
3.4	Задаю имя начальной ветки . . . . .	8
3.5	Задаю параметры autocrlf и safecrlf . . . . .	8
3.6	Генерация ssh ключа по алгоритму rsa . . . . .	8
3.7	Генерация ssh ключа по алгоритму ed25519 . . . . .	9
3.8	Генерация ключа . . . . .	10
3.9	Аккаунт на Github . . . . .	11
3.10	Вывод списка ключей . . . . .	12
3.11	Копирование ключа в буфер обмена . . . . .	12
3.12	Настройки GitHub . . . . .	12
3.13	Добавление нового PGP ключа . . . . .	13
3.14	Добавленный ключ GPG . . . . .	13
3.15	Настройка подписей Git . . . . .	13
3.16	Авторизация в gh . . . . .	14
3.17	Завершение авторизации через браузер . . . . .	14
3.18	Завершение авторизации . . . . .	15
3.19	Создание репозитория . . . . .	15
3.20	Перемещение между директориями . . . . .	16
3.21	Удаление файлов и создание каталогов . . . . .	16
3.22	Отправка файлов на сервер . . . . .	16

## **Список таблиц**

# 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

### ## Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh` (рис. 3.1).

```
root
[aksokirka@fedora ~]$ dnf install git
Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожалуйста, войдите в систему как пользо
вышенными правами или используйте опции "--assumeno" или "--downloadonly", чтобы выполнить команду без изменения сост
емы.
[aksokirka@fedora ~]$ sudo -i
[sudo] пароль для aksokirka:
[root@fedora ~]# dnf install git
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates                                100% | 81.1 KiB/s | 21.3 KiB
Fedora 41 - x86_64 - Updates                                100% | 2.0 MiB/s | 2.9 MiB
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@fedora ~]#
```

Рис. 3.1: Установка git и gh

### ## Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту(рис. 3.2).

```
root
[aksokirka@fedora ~]$ git config --global user.name "Anna Sokirka"
[aksokirka@fedora ~]$ git config --global user.email "1132246758@pfur.ru"
[aksokirka@fedora ~]$
```

Рис. 3.2: Задаю имя и email владельца репозитория

Настраиваю utf-8 в выводе сообщений git для их корректного отображения(рис. 3.3).

```
[aksokirka@fedora ~]$ git config --global core.quotePath false
[aksokirka@fedora ~]$
```

Рис. 3.3: Настройка utf-8 в выводе сообщений git

Начальной ветке задаю имя master(рис. 3.4).

```
[aksokirka@fedora ~]$ git config --global init.defaultBranch master
[aksokirka@fedora ~]$
```

Рис. 3.4: Задаю имя начальной ветки

Задаю параметры autocrlf и safecrlf для корректного отображения конца строки(рис. 3.5).

```
[aksokirka@fedora ~]$ git config --global core.autocrlf input
[aksokirka@fedora ~]$
```

Рис. 3.5: Задаю параметры autocrlf и safecrlf

## ##Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa(рис. 3.6).

```
[aksokirka@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aksokirka/.ssh/id_rsa):
Created directory '/home/aksokirka/.ssh'.
Enter passphrase for "/home/aksokirka/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aksokirka/.ssh/id_rsa
Your public key has been saved in /home/aksokirka/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:W/TLsfGIC0HbcCNw55iEtg5QSywuj3olokNx0KIUI aksokirka@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|E .. =oo o o |
|o. . .o * * . |
|. o + . o.o * |
| o = . .o.B = |
|. * o So..o= . |
| + o . o . +. |
| . . . . + .. |
| . . . . . . |
|... .. |
+----[SHA256]-----+
[aksokirka@fedora ~]$
```

Рис. 3.6: Генерация ssh ключа по алгоритму rsa



Создаю ключ ssh по алгоритму ed25519(рис. 3.7).

```
[aksokirka@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aksokirka/.ssh/id_ed25519):
Enter passphrase for "/home/aksokirka/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aksokirka/.ssh/id_ed25519
Your public key has been saved in /home/aksokirka/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:8ty8K/Ggvuv7mkX7byUAB2eZqEw1W5gBqk6GLbPodfE aksokirka@fedora
The key's randomart image is:
+--[ED25519 256]--+
|      .o=*+o      |
|      .. =*+      |
|      .o ..o      |
| o . o .          |
|+= .. S .         |
|.B   o=o+ . .     |
|o .. . E=+o o     |
|. . . .o.....    |
|. . o0*..++       |
+-----[SHA256]-----+
[aksokirka@fedora ~]$
```

Рис. 3.7: Генерация ssh ключа по алгоритму ed25519

### ##Создание ключа GPG ma

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. 3.8).

```

[aksokirka@fedora ~]$ 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 '/home/aksokirka/.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? 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: SokirkaAnna
Адрес электронной почты: 1132246758@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "SokirkaAnna <1132246758@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 

```

Рис. 3.8: Генерация ключа

```

[aksokirka@fedora ~]$ 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 red
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/aksokirka/.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? 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

Ваше полное имя: SokirkaAnna
Адрес электронной почты: 1132246758@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "SokirkaAnna <1132246758@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(

```

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

## ##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. 3.9).

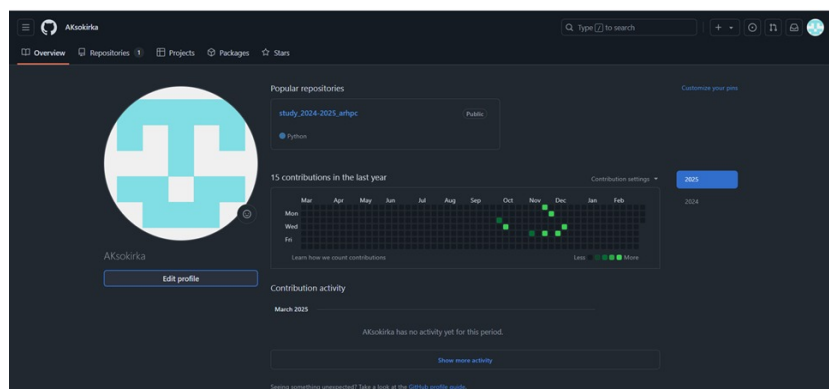


Рис. 3.9: Аккаунт на Github

## ##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. 3.10).

```
[aksokirka@fedora ~]$ 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: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/2847432B111F69E3 2025-03-04 [SC]
      76A672E03814C2C796FF446C2847432B111F69E3
uid           [ абсолютно ] SokirkaAnna <1132246758@pfur.ru>
ssb   rsa4096/6EDF4C36783AA320 2025-03-04 [E]
[aksokirka@fedora ~]$
```

Рис. 3.10: Вывод списка ключей

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip(рис. 3.11).

```
[aksokirka@fedora ~]$ gpg --armor --export 2847432B111F69E3 | xclip -sel clip
```

Рис. 3.11: Копирование ключа в буфер обмена

Открываю настройки GitHub, ищу среди них добавление GPG ключа(рис. 3.12).



Рис. 3.12: Настройки GitHub

Нажимаю на “New GPG key” и вставляю в поле ключ из буфера обмена(рис. 3.13).

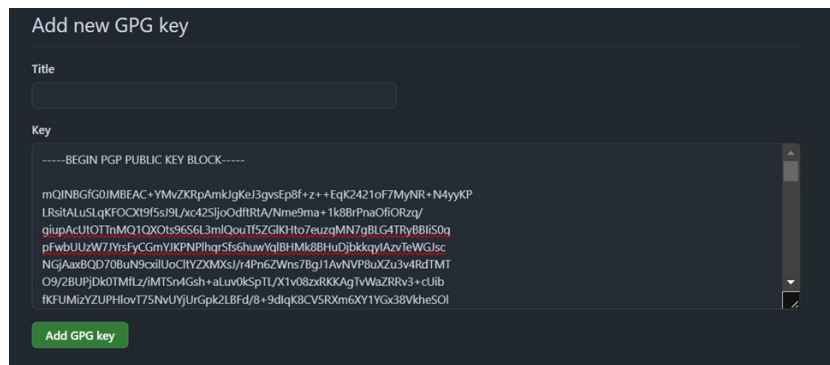


Рис. 3.13: Добавление нового PGP ключа

Я добавила ключ GPG на GitHub(рис. 3.14).

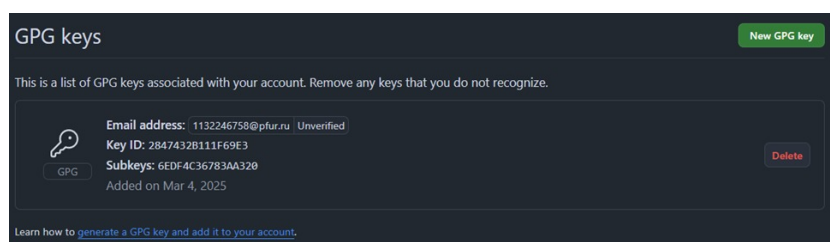


Рис. 3.14: Добавленный ключ GPG

## ##Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов(рис. 3.15).

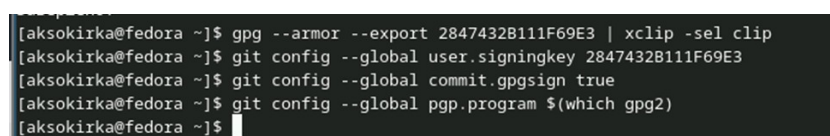


Рис. 3.15: Настройка подписей Git

## ##Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер(рис. 3.16).

```
[aksokirka@fedora ~]$ 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: 5130-13F9
Press Enter to open https://github.com/login/device in your browser...
```

Рис. 3.16: Авторизация в gh

Завершаю авторизацию на сайте(рис. 3.17).

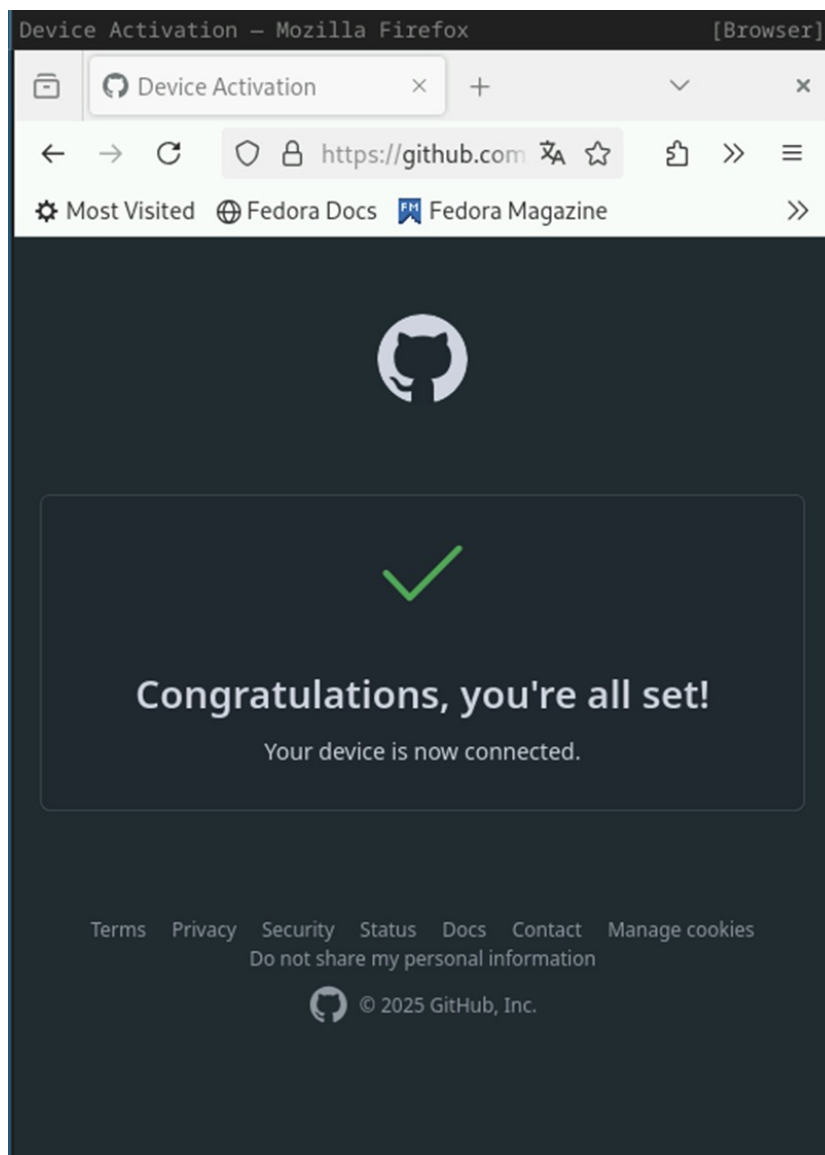


Рис. 3.17: Завершение авторизации через браузер

Вижу сообщение о завершении авторизации под именем aksokirka(рис. 3.18).

```

✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as AKsokirka
[aksokirka@fedora ~]$

```

Рис. 3.18: Завершение авторизации

## ##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=yamadharm/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. 3.19).

```

[aksokirka@fedora ~]$ cd ~/work/study/2024-2025/"Операционные системы"
[aksokirka@fedora Операционные системы]$ gh repo create study_2024-2025_os-intro --template=yamadharm/course-directory-student-template --public
✓ Created repository AKsokirka/study_2024-2025_os-intro on GitHub
https://github.com/AKsokirka/study_2024-2025_os-intro
[aksokirka@fedora Операционные системы]$ git clone --recursive https://github.com/AKsokirka/study_2024-2025_os-intro
Клонирование в «study_2024-2025_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), 19.37 КиБ | 192.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/aksokirka/work/study/2024-2025/Операционные системы/study_2024-2025_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)
Получение объектов: 100% (111/111), 102.17 КиБ | 857.00 КиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/aksokirka/work/study/2024-2025/Операционные системы/study_2024-2025_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)
Получение объектов: 100% (142/142), 341.09 КиБ | 1.13 МБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
[aksokirka@fedora Операционные системы]$

```

Рис. 3.19: Создание репозитория



Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls`(рис. 3.20).

```
[aksokirka@fedora 2024-2025]$ cd "Операционные системы"
[aksokirka@fedora Операционные системы]$ ls
study_2024-2025_os-intro
[aksokirka@fedora Операционные системы]$ cd study_2024-2025_os-intro
[aksokirka@fedora study_2024-2025_os-intro]$ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
config        LICENSE  package.json  README.git-flow.md  template
[aksokirka@fedora study_2024-2025_os-intro]$
```

Рис. 3.20: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile`(рис. 3.21).

```
[aksokirka@fedora study_2024-2025_os-intro]$ rm package.json
[aksokirka@fedora study_2024-2025_os-intro]$
```

Рис. 3.21: 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`. Отправляю файлы на сервер с помощью `git push`(рис. 3.22).

```
[aksokirka@fedora study_2024-2025_os-intro]$ echo os-intro > COURSE
[aksokirka@fedora study_2024-2025_os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule       Update submules

[aksokirka@fedora study_2024-2025_os-intro]$ git add .
[aksokirka@fedora study_2024-2025_os-intro]$ git commit -am 'feat(main): make course structure'
[main fc416a7] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
[aksokirka@fedora study_2024-2025_os-intro]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 949 байтов | 949.00 Киб/с, готово.
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 https://github.com/Aksokirka/study_2024-2025_os-intro
   b3b914f..fc416a7  master -> master
[aksokirka@fedora study_2024-2025_os-intro]$
```

Рис. 3.22: 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.

1. Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т. д. VCS применяются для: Хранения полной истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.
2. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в распределенных (децентрализо-

ванных) системах контроля версий центральный репозиторий не является обязательным.

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.
7. Создание основного дерева репозитория: `git init` 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 вет-

ки: `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 9. 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**

<https://esystem.rudn.ru/enrol/index.php?id=113>