

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

дисциплина: Архитектура компьютера

Студент: Серебрякова Дарья Ильинична

№ Студенческого билета: 1132246733

Группа: НКАбд-04-24

МОСКВА

2024 г.

Содержание

1 Цель работы	3
2 Теоретическое введение	4
2.1 Системы контроля версий. Общие понятия	4
2.2 Система контроля версий Git.....	5
3 Порядок выполнения лабораторной работы	5
3.1 Настройка github	5
3.2 Базовая настройка git.....	5
3.3 Создание SSH ключа.....	6
3.4 Создание рабочего пространства и репозитория курса на основе шаблона	7
3.5 Создание репозитория курса на основе шаблона	8
3.6 Настройка каталога курса	9
4. Вывод.....	10
Рисунок 1 Созданный на github.com аккаунт.....	5
Рисунок 2 Создание предварительной конфигурации	6
Рисунок 3 Настройка utf-8 в выводе сообщений git	6
Рисунок 4 Создание имени начальной ветки и параметров для нее.....	6
Рисунок 5 сохранение ключа.....	6
Рисунок 6 копия ключа в буфер обмена.....	7
Рисунок 7 Создание ключа	7
Рисунок 8 Ключ создан	7
Рисунок 9 Создание каталога для предмета.....	8
Рисунок 10 Создание репозитория.....	8
Рисунок 11 Переход в каталог курса	9
Рисунок 12 Ссылка для клонирования.....	9
Рисунок 13 Клонирование созданного репозитория.....	9
Рисунок 14 Переход в каталог курса	9
Рисунок 15 Удаление лишних файлов	9
Рисунок 16 Создание необходимых каталогов	10

1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git

2 Теоретическое введение

2.1 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

2.2 Система контроля версий *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

3 Порядок выполнения лабораторной работы

3.1 Настройка *github*

Создала учётную запись на сайте <https://github.com/> и заполнила основные данные. Вошла в свой аккаунт

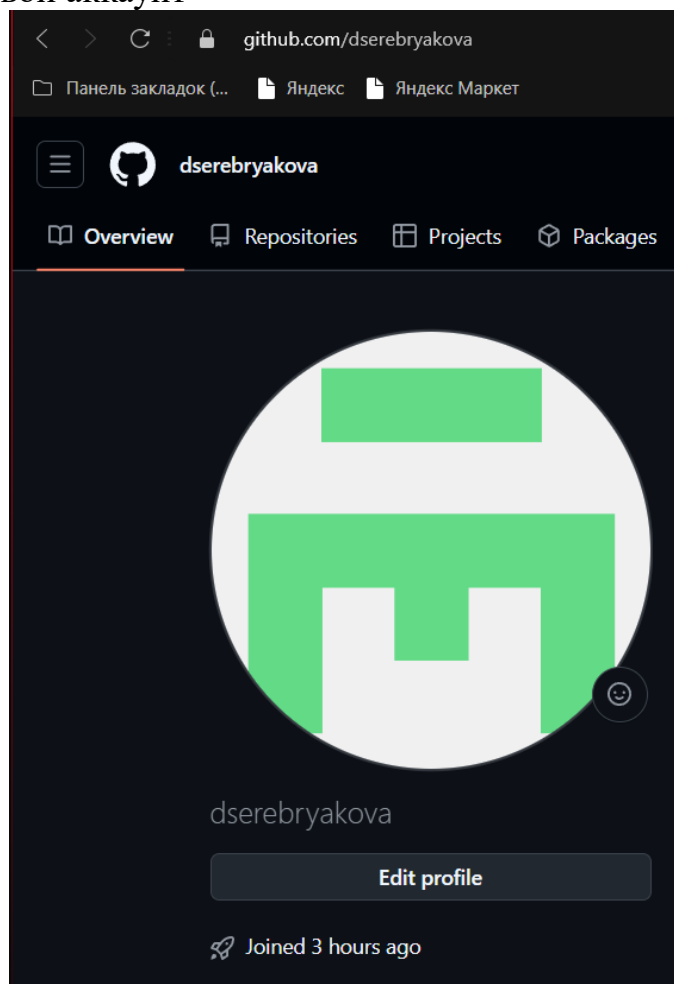


Рисунок 1 Созданный на *github.com* аккаунт

3.2 Базовая настройка *git*

Открываю виртуальную машину, захожу в терминал. Сделаю предварительную конфигурацию *git*. Ввожу команду `git config --global user.name ""`, указывая свое имя, фамилию. Далее ввожу команду `git config --global user.email "work@mail"`, указывая в ней свою электронную почту

```
diserebryakova@fedora:~$ git config --global user.name "<Дарья Серебрякова>"
diserebryakova@fedora:~$ git config --global user.email "<1132246733@pfur.ru>"
diserebryakova@fedora:~$
```

Рисунок 2 Создание предварительной конфигурации

Настраиваю utf-8 в выводе сообщений git, введя предложенную команду

```
diserebryakova@fedora:~$ git config --global core.quotePath false
diserebryakova@fedora:~$
```

Рисунок 3 Настройка utf-8 в выводе сообщений git

Задаю имя начальной ветки (буду называть её masster). Также задаю параметр autocrlf со значением input и параметр safecrlf со значением warn

```
diserebryakova@fedora:~$ git config --global init.defaultBranch masster
diserebryakova@fedora:~$ git config --global core.autocrlf input
diserebryakova@fedora:~$ git config --global core.safecrlf warn
diserebryakova@fedora:~$
```

Рисунок 4 Создание имени начальной ветки и параметров для нее

3.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая свое имя (имя владельца) и электронную почту владельца. Ключ автоматически сохранится в каталоге `~/.ssh/`

```
diserebryakova@fedora:~$ ssh-keygen -C "Дарья Серебрякова <1132246733@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/diserebryakova/.ssh/id_ed25519):
Created directory '/home/diserebryakova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/diserebryakova/.ssh/id_ed25519
Your public key has been saved in /home/diserebryakova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:qLBPT0gVHptsxCid/aGUBQ8J8dxUjfRHCBDWJqrDXa4 Дарья Серебрякова <1132246733@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|  .oB*=***+  ..  |
|  .+**@o+.o.  |
|  .X+o+  .  |
|    oo.o    .  |
|  ...o.oS    |
|    ++o.  .  |
|  .+...    |
|    o oE    |
|    .  .    |
+-----[SHA256]-----+
diserebryakova@fedora:~$
```

Рисунок 5 сохранение ключа

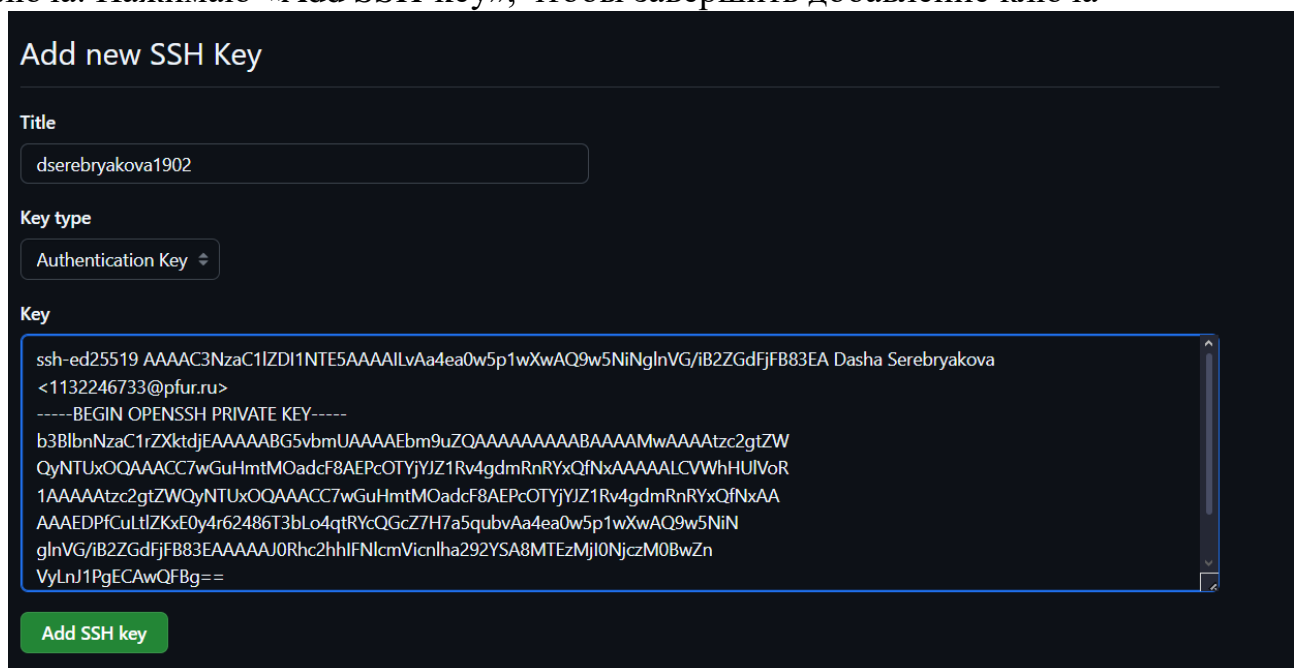
Далее необходимо загрузить сгенерённый открытый ключ. Для этого захожу на сайт <http://github.org/> под своей учётной записью и перехожу в меню Setting . После этого выбираю в боковом меню SSH and GPG keys и нажимаю кнопку New SSH key

Копирую из локальной консоли ключ в буфер обмена

```
diserebryakova@fedora:~$ ls ~/.ssh/  
id_ed25519 id_ed25519.pub  
diserebryakova@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip  
diserebryakova@fedora:~$ cat ~/.ssh/id_ed25519 | xclip -sel clip
```

Рисунок 6 копия ключа в буфер обмена

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа



Add new SSH Key

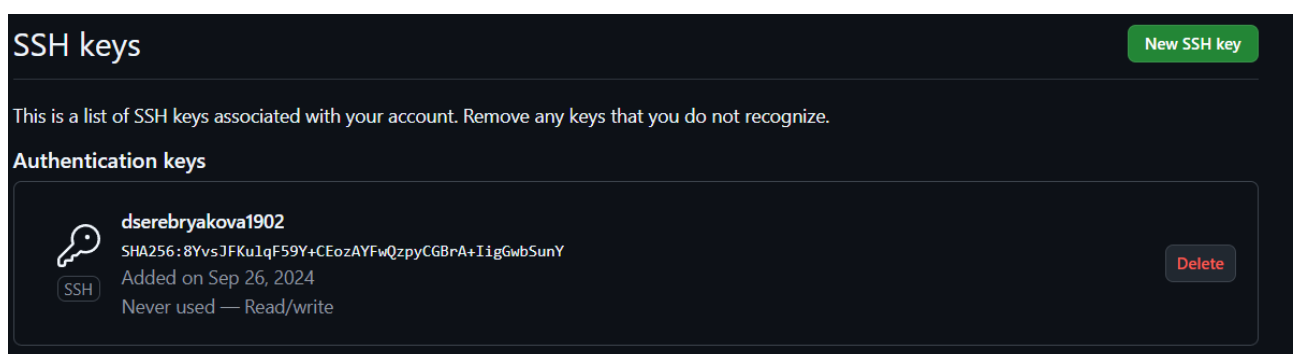
Title
dserebryakova1902

Key type
Authentication Key

Key
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIlvAa4ea0w5p1wXwAQ9w5NiNglNvG/iB2ZGdFjFB83EA Dasha Serebryakova
<1132246733@pfur.ru>
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktZjEAAAAABG5vbmUAAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAAtzc2gtZW
QyNTUxOQAAACC7wGuHmtMOadcF8AEPcOTYjYjZ1Rv4gdmRnRyXQfNxAAAAALCVWhHUIVOR
1AAAAAtzc2gtZWQyNTUxOQAAACC7wGuHmtMOadcF8AEPcOTYjYjZ1Rv4gdmRnRyXQfNxAA
AAAEDEPfcuLtlZKxE0y4r62486T3bLo4qtRYcQGcZ7H7a5qubvAa4ea0w5p1wXwAQ9w5NiN
glNvG/iB2ZGdFjFB83EAAAAAJ0Rhc2hhIHNlcmViclha292YSA8MTEzMjI0NjczM0BwZn
VyLnJ1PgECAwQFBg==

Add SSH key

Рисунок 7 Создание ключа



SSH keys New SSH key

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

Authentication keys

Key	SHA256	Added	Used	Permissions	Action
dserebryakova1902	8YvsJFKu1qF59Y+CEozAYFwQzpyCGBrA+IigGwbSunY	Added on Sep 26, 2024	Never used	Read/write	Delete

Рисунок 8 Ключ создан

3.4 Создание рабочего пространства и репозитория курса на основе шаблона

Открываю терминал и создаю каталог для предмета «Операционные

системы», используя команду `mkdir` с ключом `-p`

```
dserebryakova@fedora:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
```

Рисунок 9 Создание каталога для предмета

3.5 Создание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном курса. Далее выбираю Use this template

В открывшемся окне задаю имя репозитория (Repository name) `study_2024-2025_os-intro` и создаю репозиторий. Репозиторий создан

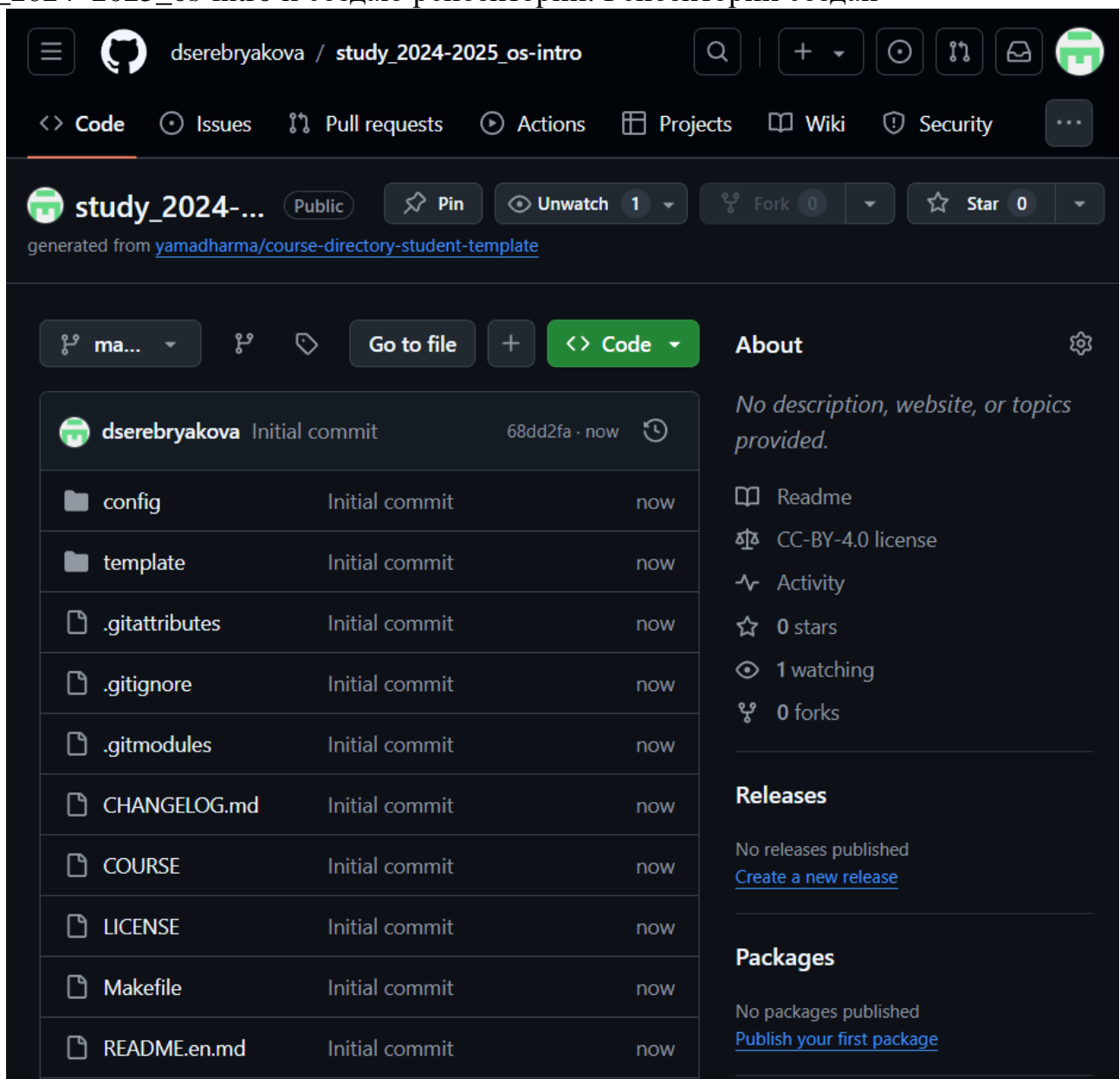


Рисунок 10 Создание репозитория

Открываю терминал и перехожу в каталог курса


```
diserebryakova@fedora:~$ cd ~/work/study/2024-2025/"Операционные системы"
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы$
```

Рисунок 11 Переход в каталог курса

Ссылку для клонирования копирую на странице созданного репозитория

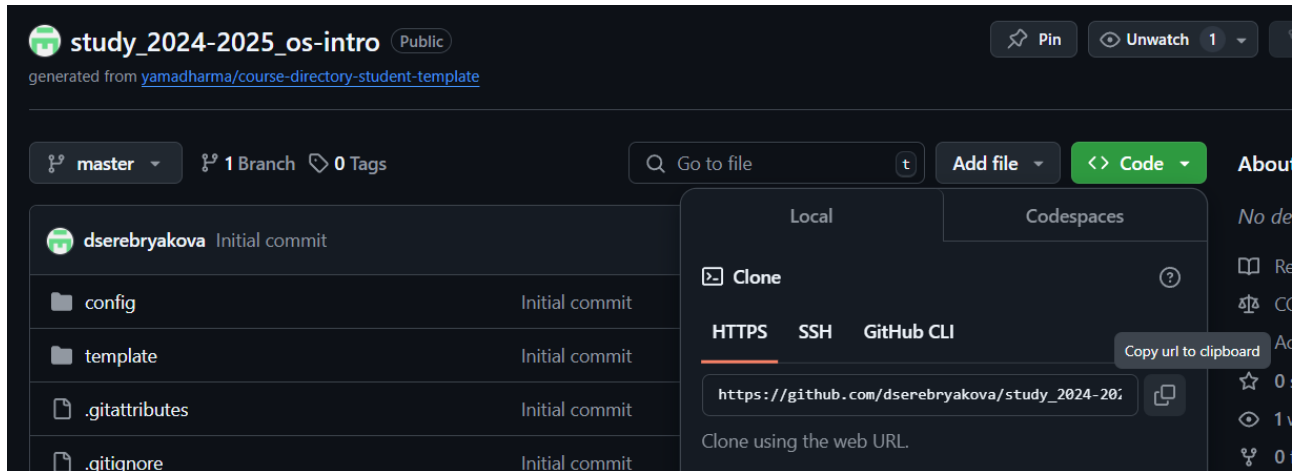


Рисунок 12 Ссылка для клонирования

Клонирую созданный репозиторий, используя только что скопированную ссылку

```
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы$ git clone --recursive
git@github.com:dserebryakova/study_2024-2025_os-intro.git os-intro
fatal: целевой путь «os-intro» уже существует и не является пустым каталогом.
```

Рисунок 13 Клонирование созданного репозитория

3.6 Настройка каталога курса

Перехожу в каталог курса командой cd

```
diserebryakova@fedora:~$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы/os-intro$
```

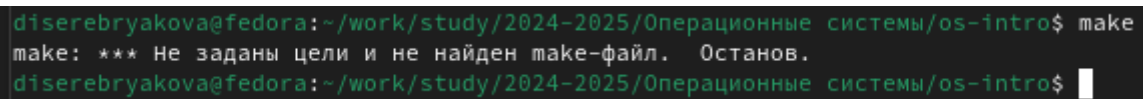
Рисунок 14 Переход в каталог курса

Пробую удалить лишние файлы командой rm

```
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.js
on
rm: невозможно удалить 'package.json': Нет такого файла или каталога
```

Рисунок 15 Удаление лишних файлов

Для создания необходимых каталогов использую команду make

A terminal window with a dark background and green text. The prompt is 'diserebryakova@fedora:~/work/study/2024-2025/Операционные системы/os-intro\$'. The command 'make' has been entered. The output is 'make: *** Не заданы цели и не найден make-файл. Останов.' followed by a new prompt line.

```
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ make
make: *** Не заданы цели и не найден make-файл. Останов.
diserebryakova@fedora:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рисунок 16 Создание необходимых каталогов

4. Вывод

В ходе выполнения работы изучены идеология и применение средств контроля версий. Приобретены практические навыки по работе с системой git