

# **Лабораторная работа № 4.**

## **Продвинутое использование git.**

**Отчет**

Анна Александровна Глущенок

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Выполнение лабораторной работы</b>	<b>7</b>
<b>4 Выводы</b>	<b>24</b>
<b>Список литературы</b>	<b>25</b>

# Список иллюстраций

3.1 Установка git-flow . . . . .	7
3.2 Установка Node.js . . . . .	8
3.3 Настройка Node.js . . . . .	8
3.4 Общепринятые коммиты . . . . .	9
3.5 Создание репозитория . . . . .	10
3.6 Первый коммит (1) . . . . .	11
3.7 Первый коммит (2) . . . . .	12
3.8 Конфигурация пакетов . . . . .	12
3.9 Заполнение параметров . . . . .	13
3.10 Добавление файлов, коммит . . . . .	14
3.11 Отправка файлов на GitHub . . . . .	14
3.12 Инициализация git flow; Префикс v . . . . .	15
3.13 Загрузка репозитория в хранилище . . . . .	16
3.14 Создание релиза, журнала изменений . . . . .	17
3.15 Добавление журнала изменений в индекс . . . . .	17
3.16 Отправка данных, создание релиза . . . . .	18
3.17 Объединение веток . . . . .	19
3.18 Создание релиза . . . . .	20
3.19 Создание журнала изменений . . . . .	20
3.20 Добавление релизной ветки в основную . . . . .	21
3.21 Отправка данных, создание релиза . . . . .	22
3.22 Проверка наличия файлов . . . . .	22
3.23 Проверка релизов . . . . .	23

# **Список таблиц**

# **1 Цель работы**

Получение навыков правильной работы с репозиториями git.

## **2 Задание**

1. Выполнить работу для тестового репозитория.
2. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

### 3 Выполнение лабораторной работы

Установим git-flow из коллекции репозиториев Copr:

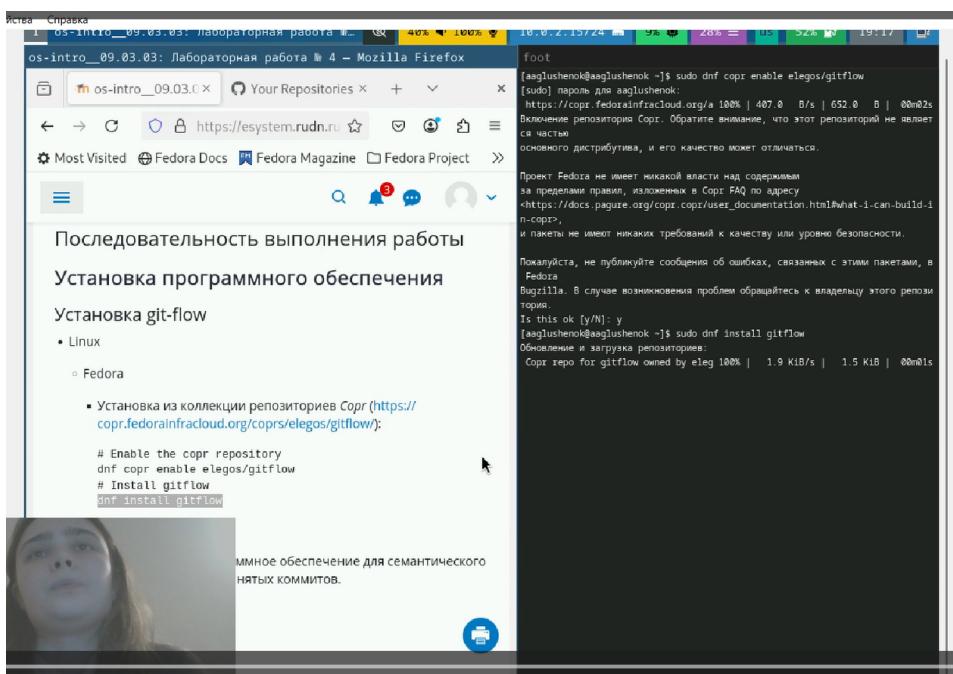


Рис. 3.1: Установка git-flow

Установим Node.js для Fedora:

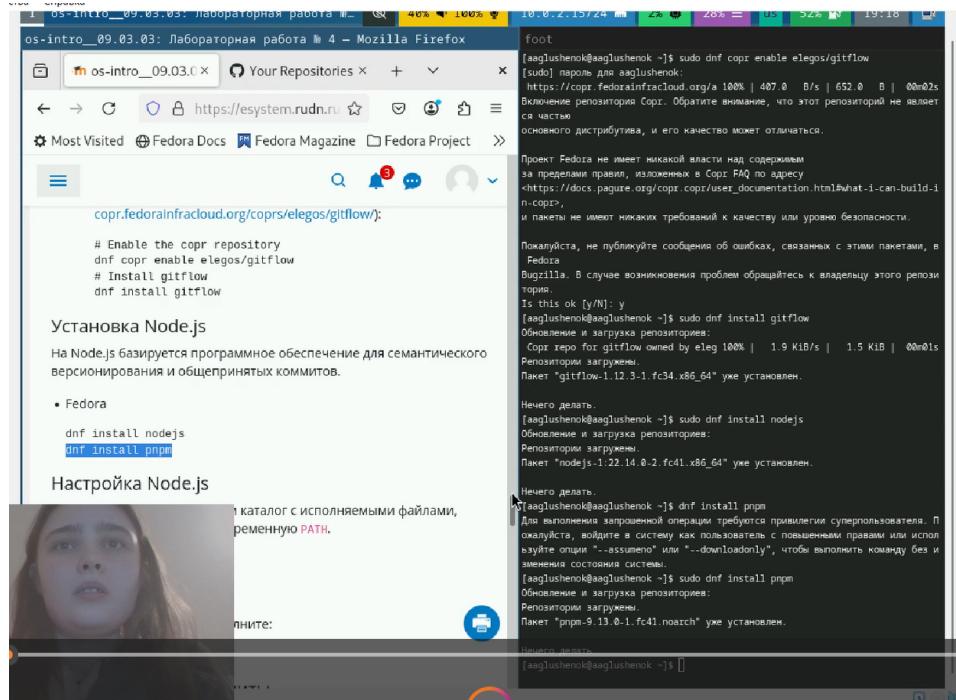


Рис. 3.2: Установка Node.js

Настроим Node.js - добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH;

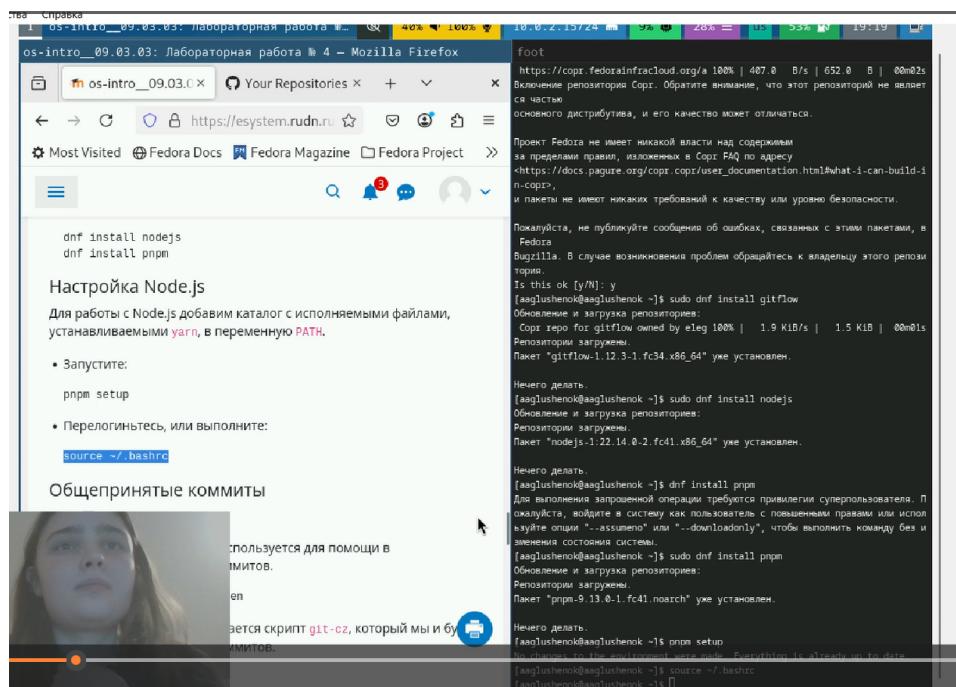


Рис. 3.3: Настройка Node.js

Общепринятые коммиты: введем команды из текста лабораторной работы для помощи в формировании коммитов, для помощи в создании логов:

The screenshot shows a Firefox browser window with a terminal-like interface on the right side. The terminal window displays a command-line session with several commands entered and their outputs. The top part of the terminal shows the command `npm add -g standard-changelog`. Below it, a commit message for "git-extended" is shown, followed by a series of npm install commands for Node.js packages like `glob@7.2.3` and `inflight@1.0.6`. The bottom part of the terminal shows the command `git commit -m "first commit"` and its output.

```
Is this ok [y/N]: y
[aaglushenok@aaglushenok ~]$ sudo dnf install gitflow
Обновление и загрузка репозиториев:
Скопиhero для gitflow owned by ele6 100% | 1.9 Kib/s | 1.5 Kib | 00ms
Репозитории загружены.
Пакет "gitflow-1.12.3-1.fc34.x86_64" уже установлен.

Ничего делать.
[aaglushenok@aaglushenok ~]$ sudo dnf install nodejs
Обновление и загрузка репозиториев:
Репозитории загружены.
Пакет "nodejs-12.14.0-2.fc41.x86_64" уже установлен.

Ничего делать.
[aaglushenok@aaglushenok ~]$ dnf install pnpm
Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожалуйста, войдите в систему как пользователь с повышенными правами или используйте опцию "--assumeno" или"--downloadonly", чтобы выполнить команду без изменения состояния системы.
[aaglushenok@aaglushenok ~]$ sudo dnf install pnpm
Обновление и загрузка репозиториев:
Репозитории загружены.
Пакет "pnpm-9.13.0-1.fc41.noarch" уже установлен.

Ничего делать.
[aaglushenok@aaglushenok ~]$ pnpm setup
No changes to the environment were made. Everything is already up to date.
[aaglushenok@aaglushenok ~]$ source ~/.bashrc
[aaglushenok@aaglushenok ~]$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 5.3s
[aaglushenok@aaglushenok ~]$ pnpm add -g standard-changelog
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 4.6s
[aaglushenok@aaglushenok ~]$ 
```

Рис. 3.4: Общепринятые коммиты

Создадим новый репозиторий на GitHub, назовем его git-extended

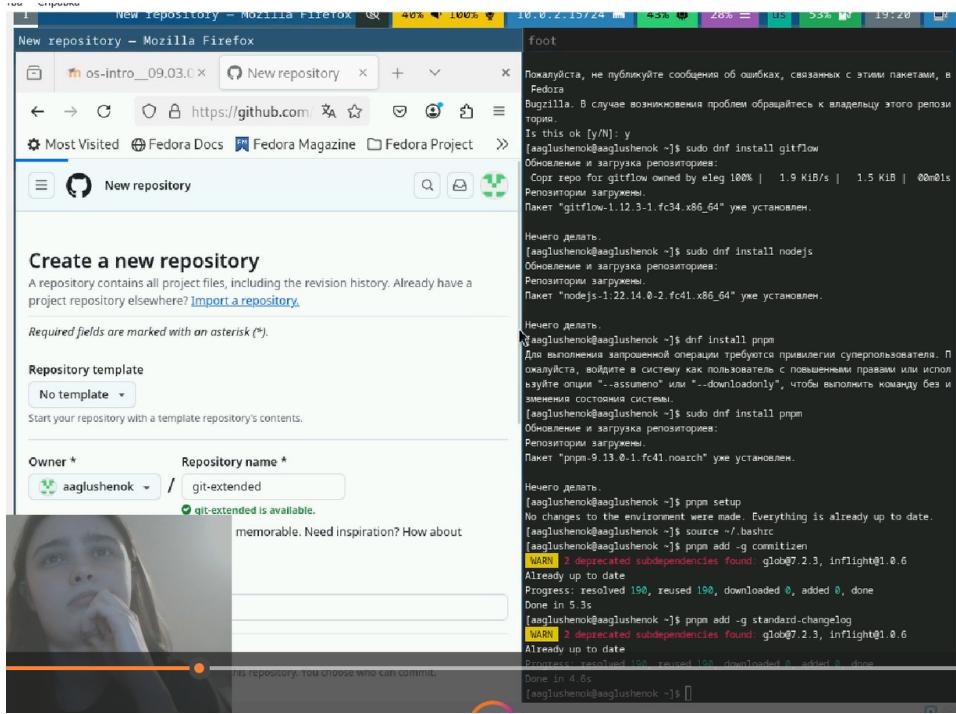


Рис. 3.5: Создание репозитория

Рекурсивно клонируем созданный репозиторий. Перейдем в него. Создадим в репозитории файл README.md, запишем в файл слово README, для того чтобы не работать с пустым репозиторием:

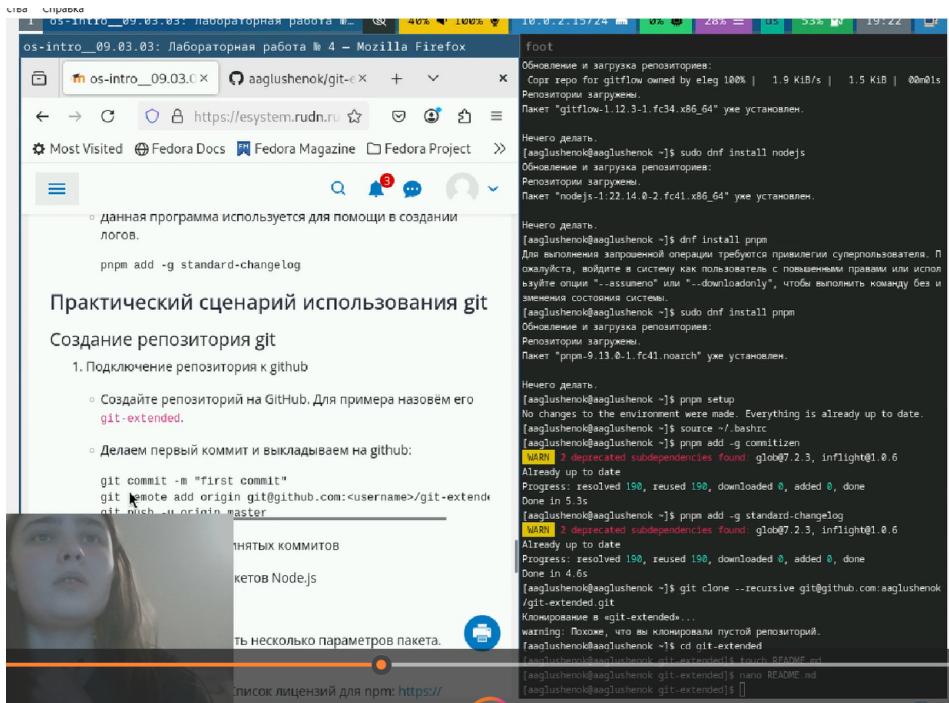


Рис. 3.6: Первый коммит (1)

Пропишем команду `git add .`, после чего вставим в терминал команды из текста лабораторной работы для первого коммита и отправки файлов:

```

os-intro_09.03.03: Лабораторная работа 4 – Mozilla Firefox
https://esystem.rudn.ru
Most Visited Fedora Docs Fedora Magazine Fedora Project
pnpm add -g standard-changelog

foot
[aa@lushmanok@aa@lushmanok ~]$ pnpm setup
[aa@lushmanok@aa@lushmanok ~]$ source ~/.bashrc
[aa@lushmanok@aa@lushmanok ~]$ pnpm add -g commitizen
[aa@lushmanok@aa@lushmanok ~]$ git clone --recursive git@github.com:aaglushenok/git-extended.git
Клонирование в 'git-extended'...
warning: Пожале, что мы клонировали пустой репозиторий.
[aa@lushmanok@aa@lushmanok ~]$ cd git-extended
[aa@lushmanok@aa@lushmanok git-extended]$ touch README.md
[aa@lushmanok@aa@lushmanok git-extended]$ nano README.md
[aa@lushmanok@aa@lushmanok git-extended]$ git add .
[aa@lushmanok@aa@lushmanok git-extended]$ git commit -m "first commit"
[main (новая ветвь) 13f7cd2] first commit
 1 file changed, 1 insertion(+)
create mode 100644 README.md
[aa@lushmanok@aa@lushmanok git-extended]$ git remote add origin git@github.com:aaglushenok
aa@lushmanok@aa@lushmanok git-extended.git
error: внешний репозиторий 'origin' уже существует
[aa@lushmanok@aa@lushmanok git-extended]$ git push -u origin main
Перечислено объектов: 3, готово.
Подпись объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 874 байта | 874.00 Кб/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aaglushenok/git-extended.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
[aa@lushmanok@aa@lushmanok git-extended]$ 

```

Рис. 3.7: Первый коммит (2)

Конфигурация общепринятых коммитов, для пакетов Node.js:

```

os-intro_09.03.03: Лабораторная работа 4 – Mozilla Firefox
https://esystem.rudn.ru
Most Visited Fedora Docs Fedora Magazine Fedora Project
git push -u origin master

foot
[aa@lushmanok@aa@lushmanok ~]$ pnpm add -g standard-changelog
[aa@lushmanok@aa@lushmanok ~]$ git clone --recursive git@github.com:aaglushenok/git-extended.git
Клонирование в 'git-extended'...
warning: Пожале, что мы клонировали пустой репозиторий.
[aa@lushmanok@aa@lushmanok ~]$ cd git-extended
[aa@lushmanok@aa@lushmanok git-extended]$ touch README.md
[aa@lushmanok@aa@lushmanok git-extended]$ nano README.md
[aa@lushmanok@aa@lushmanok git-extended]$ git add .
[aa@lushmanok@aa@lushmanok git-extended]$ git commit -m "first commit"
[main (новая ветвь) 13f7cd2] first commit
 1 file changed, 1 insertion(+)
create mode 100644 README.md
[aa@lushmanok@aa@lushmanok git-extended]$ git remote add origin git@github.com:aaglushenok
aa@lushmanok@aa@lushmanok git-extended.git
error: внешний репозиторий 'origin' уже существует
[aa@lushmanok@aa@lushmanok git-extended]$ git push -u origin main
Перечислено объектов: 3, готово.
Подпись объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 874 байта | 874.00 Кб/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aaglushenok/git-extended.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
[aa@lushmanok@aa@lushmanok git-extended]$ pnpm init
Wrote to /home/aaglushenok/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" & exit 1"
  },
  "keywords": [],
  "author": ""
}

package.json приобретает вид:
ended".
aa@lushmanok@aa@lushmanok git-extended$ nano package.json
[aa@lushmanok@aa@lushmanok git-extended]$ nano package.json

```

Рис. 3.8: Конфигурация пакетов

Заполним параметры конфигурационного пакета:

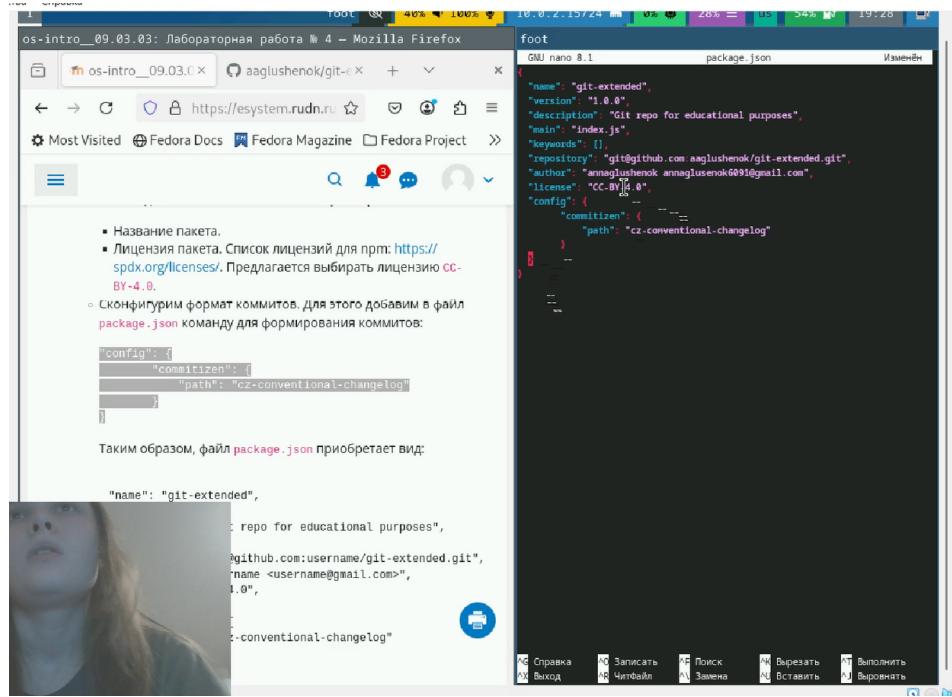


Рис. 3.9: Заполнение параметров

Добавим новые файлы, выполним коммит, выбирая в окне терминала нужные настройки:

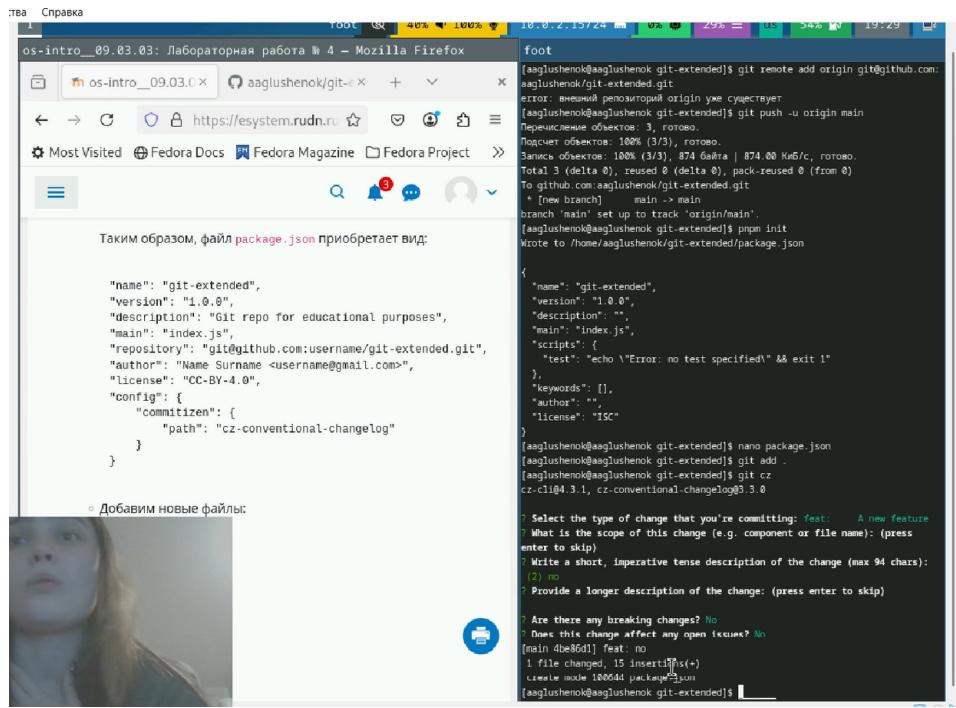


Рис. 3.10: Добавление файлов, коммит

## Отправим файлы на github::

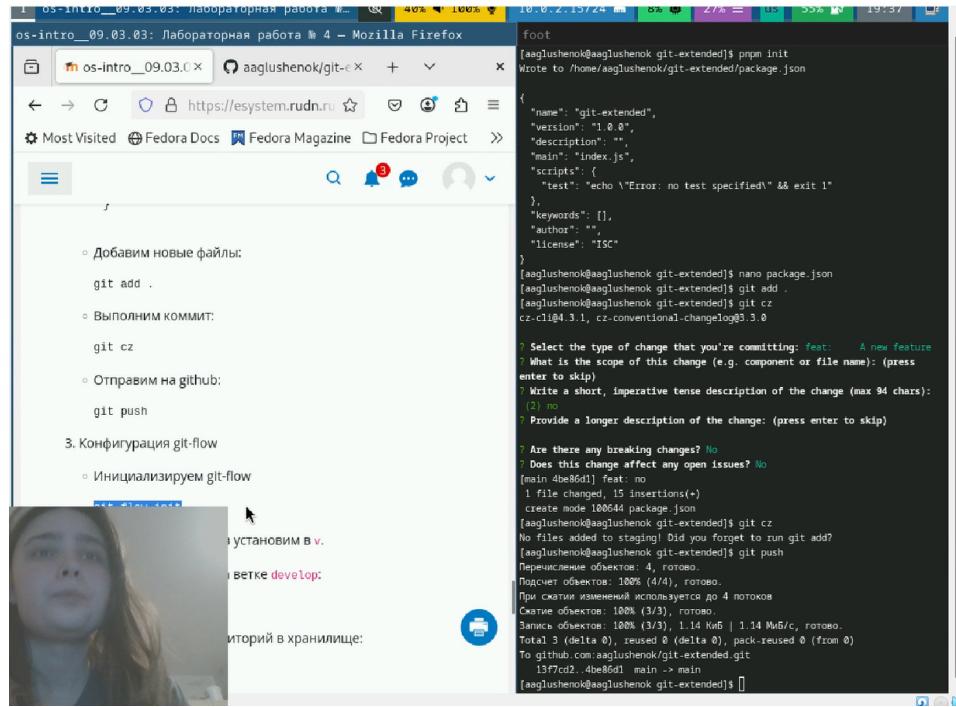
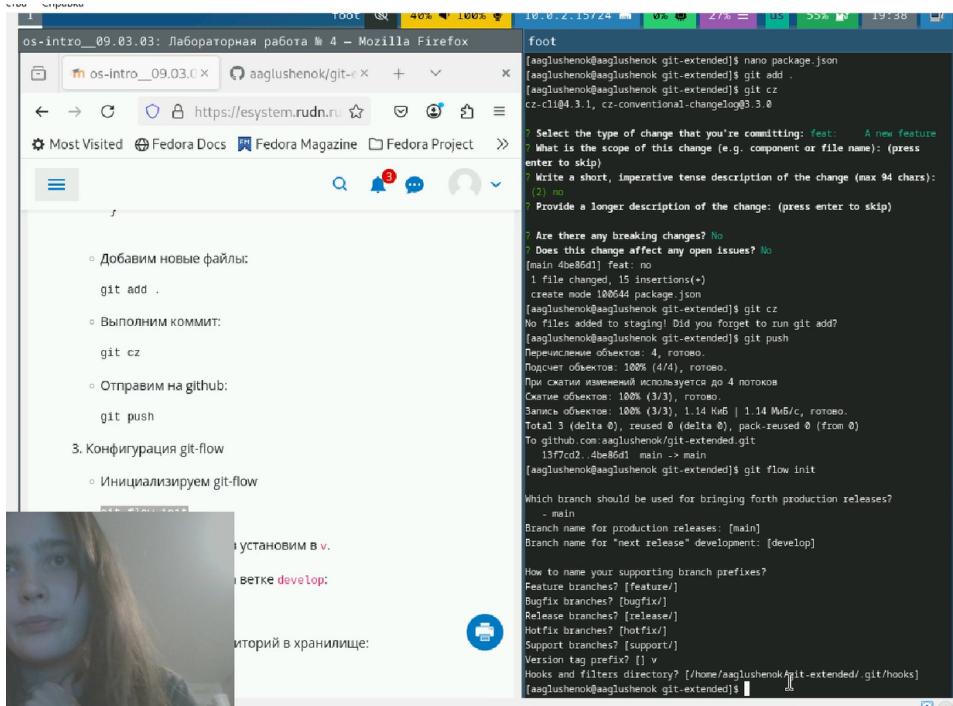


Рис. 3.11: Отправка файлов на GitHub

Конфигурация git-flow. Инициализируем git-flow, выбирая в окне терминала необходимые параметры. Установим префикс для ярлыков в v::



The screenshot shows a terminal window titled "os-intro\_09.03.03: Лабораторная работа № 4 – Mozilla Firefox". The terminal is running a script to initialize git-flow. The script asks several questions:

- Select the type of change that you're committing: **feat: A new feature**
- What is the scope of this change (e.g. component or file name): **(press enter to skip)**
- Write a short, imperative tense description of the change (max 94 chars): **(2) no**
- Provide a longer description of the change: **(press enter to skip)**
- Are there any breaking changes? **No**
- Does this change affect any open issues? **No**
- 1 file changed, 15 insertions(+)  
create mode 100644 package.json
- [aaglushenok@aaglushenok git-extended]\$ git cz  
No files added to staging! Did you forget to run git add?  
[aaglushenok@aaglushenok git-extended]\$ git push  
Перенесение объектов: 4, готово.  
Подсчет объектов: 100% (4/4), готово.  
При сканировании изменений используется до 4 потоков  
Сканирование объектов: 100% (3/3), готово  
Запись объектов: 100% (3/3), 1.14 Кб | 1.14 Мб/с, готово.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:aaglushenok/git-extended.git  
 15f7cd2..a4be86d1 main -> main
- [aaglushenok@aaglushenok git-extended]\$ git flow init
- Which branch should be used for bringing forth production releases?  
- main
- Branch name for production releases: **[main]**
- Branch name for "next release" development: **[develop]**
- How to name your supporting branch prefixes?  
Feature branches? **[feature/]**  
Bugfix branches? **[bugfix/]**  
Release branches? **[release/]**  
Hotfix branches? **[hotfix/]**  
Support branches? **[support/]**  
Version tag prefix? **[] v**  
Hooks and filters directory? **[/home/aaglushenok/.git/hooks]**
- [aaglushenok@aaglushenok git-extended]\$

Рис. 3.12: Инициализация git flow; Префикс v

Проверим, что мы на ветке develop. Загрузим весь репозиторий в хранилище:

```

[main 4be86d1] feat: no
 1 file changed, 15 insertions(+)
 create mode 100644 package.json
[aaglushenok@aaglushenok git-extended]$ git cz
No files added to staging! Did you forget to run git add?
[aaglushenok@aaglushenok git-extended]$ git push
Перенесение объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сканировании используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.14 KiB | 1.14 MiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aaglushenok/git-extended.git
  13f7cd2..4be86d1  main -> main
[aaglushenok@aaglushenok git-extended]$ git flow init

Which branch should be used for bringing forth production releases?
  - main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v

Hooks and filters directory? [/home/aaglushenok/.git/hooks]
[aaglushenok@aaglushenok git-extended]$ git branch
* develop
  main
[aaglushenok@aaglushenok git-extended]$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/aaglushenok/git-extended/pull/new/develop
remote:
To github.com:aaglushenok/git-extended.git
 * [new branch]      develop -> develop
[aaglushenok@aaglushenok git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[aaglushenok@aaglushenok git-extended]$ 

```

Отправим на github:  
git push

3. Конфигурация git-flow

- Инициализируем git-flow  
git flow init
- Префикс для ярлыков установим в v.  
git branch
- Загрузите весь репозиторий в хранилище:

Ветку как вышестоящую для этой ветки:  
stream-to=origin/develop develop

версии 1.0.0  
rt 1.0.0  
изменений

Рис. 3.13: Загрузка репозитория в хранилище

Установим внешнюю ветку как вышестоящую для этой ветки. Создадим релиз с версией 1.0.0. Создадим журнал изменений:

```

os-intro_09.03.03: Лабораторная работа № 4 – Mozilla Firefox
https://esystem.rudn.ru
git push --all
git branch --set-upstream-to=origin/develop develop
git flow release start 1.0.0
standard-changelog --first-release
git add CHANGELOG.md
git commit -am 'chore(site): add changelog'
git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Create a pull request for 'develop' on GitHub by visiting:
remote: https://github.com/aaglushenok/git-extended/pull/new/develop
remote:
To github.com:aaglushenok/git-extended.git
 * [new branch]      develop -> develop
[ааглушкин@ааглушкинок git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[ааглушкин@ааглушкинок git-extended]$ git flow release start 1.0.0
Переключился на новую ветку «release/1.0.0»
Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

git flow release finish '1.0.0'

[ааглушкин@ааглушкинок git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[ааглушкин@ааглушкинок git-extended]$ git add CHANGELOG.md
[ааглушкин@ааглушкинок git-extended]$
```

Рис. 3.14: Создание релиза, журнала изменений

Добавим журнал изменений в индекс:

```

os-intro_09.03.03: Лабораторная работа № 4 – Mozilla Firefox
https://esystem.rudn.ru
git push --all
git branch --set-upstream-to=origin/develop develop
git flow release start 1.0.0
standard-changelog --first-release
git add CHANGELOG.md
git commit -am 'chore(site): add changelog'
git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Create a pull request for 'develop' on GitHub by visiting:
remote: https://github.com/aaglushenok/git-extended/pull/new/develop
remote:
To github.com:aaglushenok/git-extended.git
 * [new branch]      develop -> develop
[ааглушкин@ааглушкинок git-extended]$
```

GNU nano 8.1 /home/aaglushenok/.git/TAG\_EDITMSG Изменение

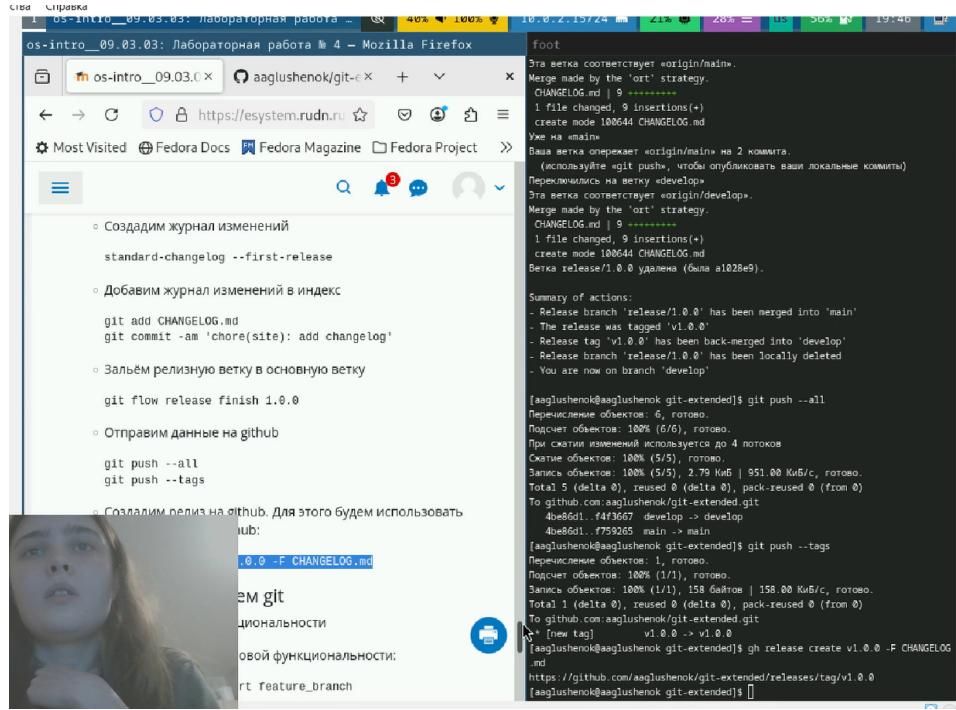
Write a message for tag:  
v1.0.0  
Lines starting with '#' will be ignored.

Сохранить измененный буфер?

Да Нет Отмена

Рис. 3.15: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку. Отправим данные на github. Создадим релиз на github. Для этого будем использовать утилиты работы с github:



```
git flow release finish 1.0.0
git push --all
git push --tags

[aa@aa-OptiPlex-5090: ~] $ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/aa@aa-OptiPlex-5090/git-extended/releases/tag/v1.0.0
[aa@aa-OptiPlex-5090: ~] $
```

Рис. 3.16: Отправка данных, создание релиза

Создадим ветку для новой функциональности. По окончании разработки новой функциональности следующим шагом следует объединить ветку `feature_branch` с `develop`:

```

os-intro_09.03.03: Лабораторная работа № 4 – Mozilla Firefox
os-intro_09.03.03 | aaglushenok/git-extended | + | x
https://esystem.rudn.ru
Most Visited Fedora Docs Fedora Magazine Fedora Project
☰ 🔍 📢 🗃

Работа с репозиторием git
1. Разработка новой функциональности
  ○ Создадим ветку для новой функциональности:
    git flow feature start feature_branch
  ○ Далее, продолжаем работу с git как обычно.
  ○ По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop:
    git flow feature finish feature_branch

2. Создание релиза gitflow
  ○ Создадим релиз с версией 1.2.3:
    rt 1.2.3
    Сии в файле package.json. Установите её
    венений
    венений в индекс

[aeaglushenok@aeaglushenok git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сканировании изменений используется до 4 потоков
Сканирование объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.79 Кб | 551.00 Кб/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aaglushenok/git-extended.git
  4be86d1...f4f3667  develop -> develop
  4be86d1...f759265  main -> main
[aeaglushenok@aeaglushenok git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 158 байтов | 158.00 Кб/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aaglushenok/git-extended.git
  * [new tag]           v1.0.0 -> v1.0.0
[aeaglushenok@aeaglushenok git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/aaglushenok/git-extended/releases/tag/v1.0.0
[aeaglushenok@aeaglushenok git-extended]$ git flow feature start feature_branch
Переключились на новую ветку «feature/feature_branch».

Summary of actions:
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:
  git flow feature finish feature_branch

[aeaglushenok@aeaglushenok git-extended]$ git flow feature finish feature_branch
Переключились на ветку «develop».
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была f4f3667).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

[aeaglushenok@aeaglushenok git-extended]$ []

```

Рис. 3.17: Объединение веток

Создадим релиз с версией 1.2.3. Обновим номер версии в файле package.json.

Установим её в 1.2.3.:

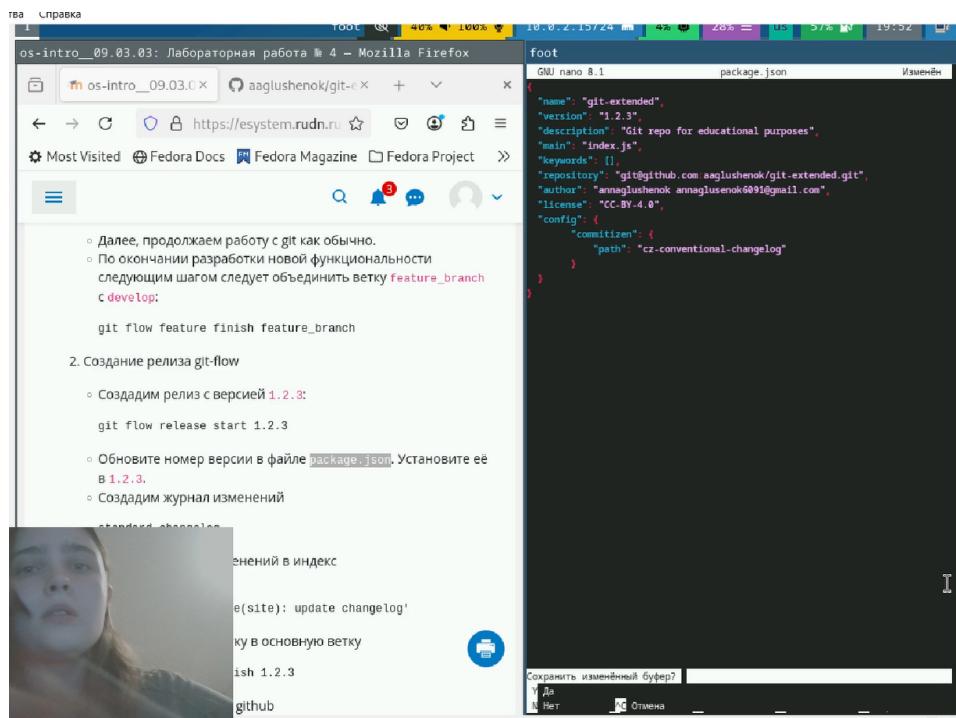


Рис. 3.18: Создание релиза

Создадим журнал изменений. Добавим журнал изменений в индекс:

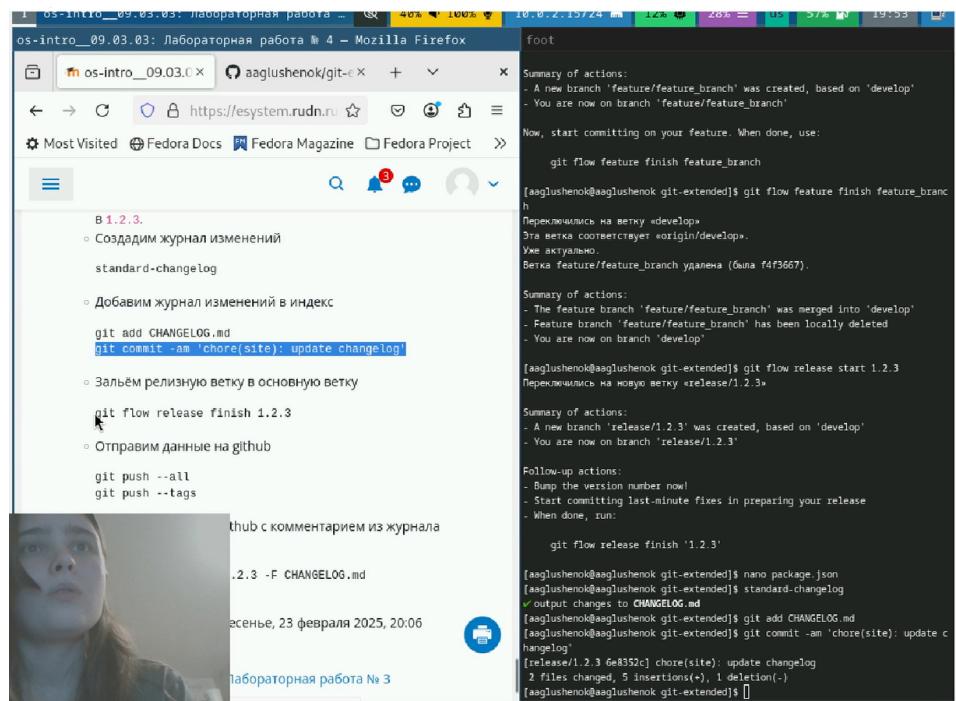


Рис. 3.19: Создание журнала изменений

Зальём релизную ветку в основную ветку, с указанием нужных параметров:

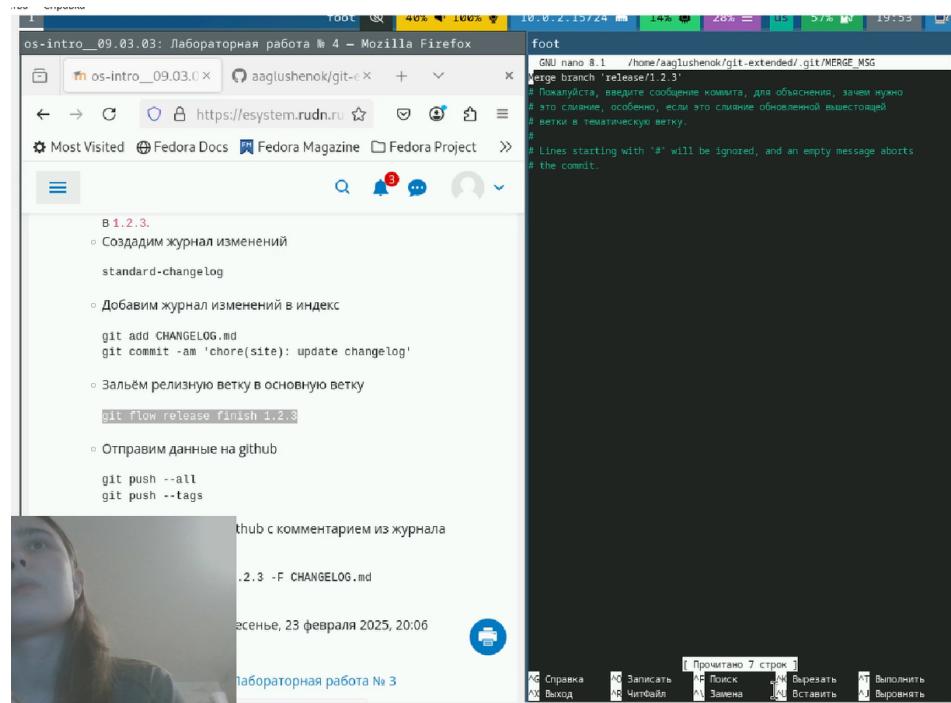


Рис. 3.20: Добавление релизной ветки в основную

Отправим данные на github. Создадим релиз на github с комментарием из журнала изменений:

```

os-intro_09.03.03: Лабораторная работа № 4 – Mozilla Firefox
https://esystem.rudn.ru
git add CHANGELOG.md
git commit -am 'chore(site): update changelog'
git flow release finish 1.2.3
git push --all
git push --tags

hub с комментарием из журнала
2.3 -F CHANGELOG.md
осень, 23 февраля 2025, 20:06
Лабораторная работа № 3

[aaaglushenok@aaaglushenok git-extended]$ git push --all
[aaaglushenok@aaaglushenok git-extended]$ git push --tags
[aaaglushenok@aaaglushenok git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
[aaaglushenok@aaaglushenok git-extended]$ https://github.com/aaaglushenok/git-extended/releases/tag/v1.2.3
[aaaglushenok@aaaglushenok git-extended]$ 

Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +---
package.json | 2 ++
2 files changed, 5 insertions(+), 1 deletion(-)
Уже на 'main'
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop».
Мердж сделан стратегией «ort».
CHANGELOG.md | 4 +---
package.json | 2 ++
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (байт 6e8352c).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

[aaaglushenok@aaaglushenok git-extended]$ git push --all
[aaaglushenok@aaaglushenok git-extended]$ git push --tags
[aaaglushenok@aaaglushenok git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
[aaaglushenok@aaaglushenok git-extended]$ https://github.com/aaaglushenok/git-extended/releases/tag/v1.2.3
[aaaglushenok@aaaglushenok git-extended]$ 

```

Рис. 3.21: Отправка данных, создание релиза

Перейдем на сайт GitHub. Убедимся, что все необходимые файлы успешно созданы:

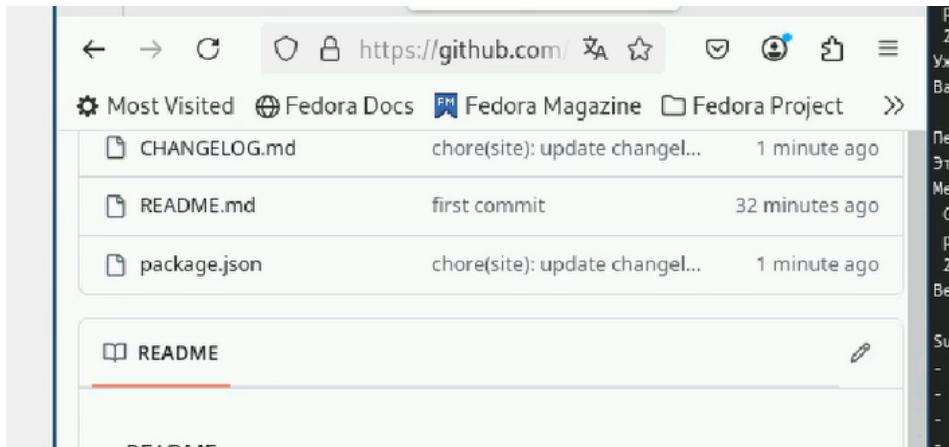


Рис. 3.22: Проверка наличия файлов

Убедимся, что все релизы созданы, и номера их версий соответствуют номерам из текста лабораторной работы:

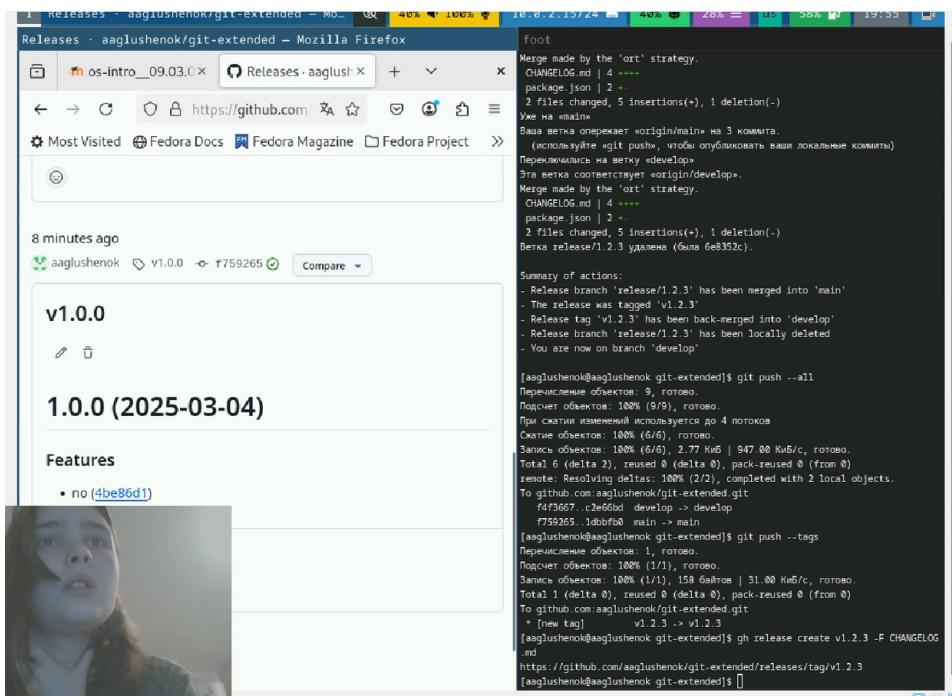


Рис. 3.23: Проверка релизов

## **4 Выводы**

В ходе выполнения лабораторной работы №4 мне удалось приобрести практические навыки правильной работы с репозиториями git.

## **Список литературы**