

# **Отчет по лаораторной работе №4**

## **Архитектура системы**

Михальский кирилл Алексеевич

# **Содержание**

|   |           |
|---|-----------|
| <b>1 Цель работы</b>                    | <b>5</b>  |
| <b>2 Задание</b>                        | <b>6</b>  |
| <b>3 Теоретическое введение</b>         | <b>7</b>  |
| <b>4 Выполнение лабораторной работы</b> | <b>8</b>  |
| <b>5 Выводы</b>                         | <b>16</b> |
| <b>Список литературы</b>                | <b>17</b> |

# **Список иллюстраций**

|                             |    |
|-----------------------------|----|
| 4.1 установка1 . . . . .    | 8  |
| 4.2 установка2 . . . . .    | 9  |
| 4.3 установка3 . . . . .    | 9  |
| 4.4 установка4 . . . . .    | 10 |
| 4.5 установка5 . . . . .    | 10 |
| 4.6 коммит1 . . . . .       | 11 |
| 4.7 конфиг . . . . .        | 11 |
| 4.8 коммит . . . . .        | 12 |
| 4.9 инициализация . . . . . | 12 |
| 4.10 проверка . . . . .     | 13 |
| 4.11 процесс . . . . .      | 13 |
| 4.12 процесс . . . . .      | 14 |
| 4.13 процесс . . . . .      | 14 |
| 4.14 завершение . . . . .   | 15 |

# **Список таблиц**

# **1 Цель работы**

Получение навыков правильной работы с репозиториями git.

## **2 Задание**

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

## 3 Теоретическое введение

Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска патчей.

Данная модель отлично подходит для организации рабочего процесса на основе релизов.

Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рамках релиза.

Последовательность действий при работе по модели Gitflow:

Из ветки `master` создаётся ветка `develop`.

Из ветки `develop` создаётся ветка `release`.

Из ветки `develop` создаются ветки `feature`.

Когда работа над веткой `feature` завершена, она слиивается с веткой `develop`.

Когда работа над веткой релиза `release` завершена, она слиивается в ветки `develop` и `release`.

Если в `master` обнаружена проблема, из `master` создаётся ветка `hotfix`.

Когда работа над веткой исправления `hotfix` завершена, она слиивается в ветки `develop` и `release`.

# 4 Выполнение лабораторной работы

## 1. Установил git-flow & gh (рис. 4.1).

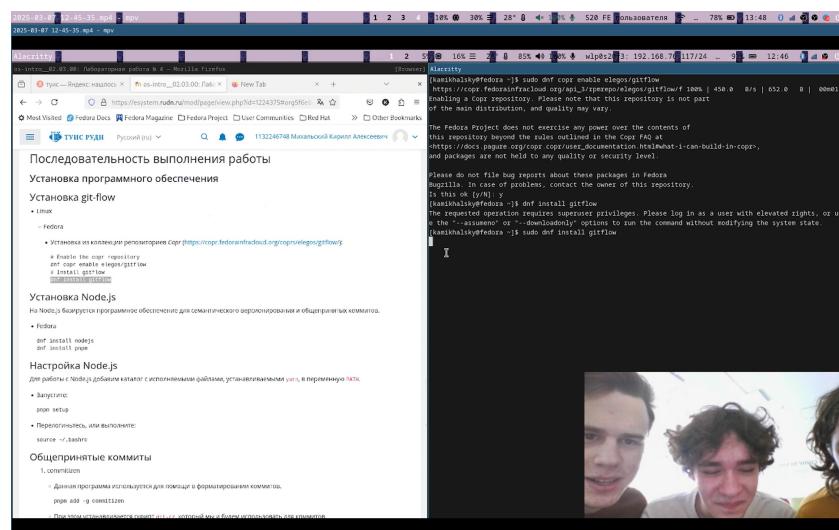


Рис. 4.1: установка1

## Установил node-js & pnpm (рис. 4.2).

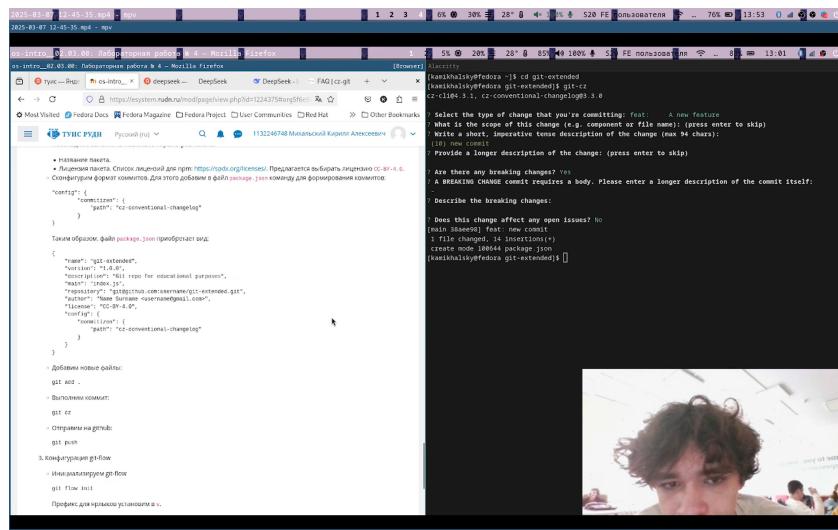


Рис. 4.2: установка2

Настроил node-js командой source ~/.bash (рис. 4.3).

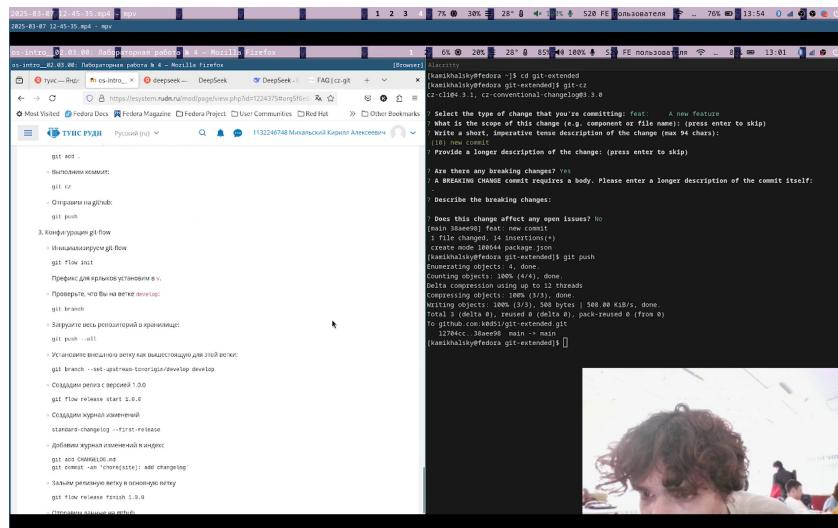


Рис. 4.3: установка3

Установил commitizen & standard-changelog 4.4).

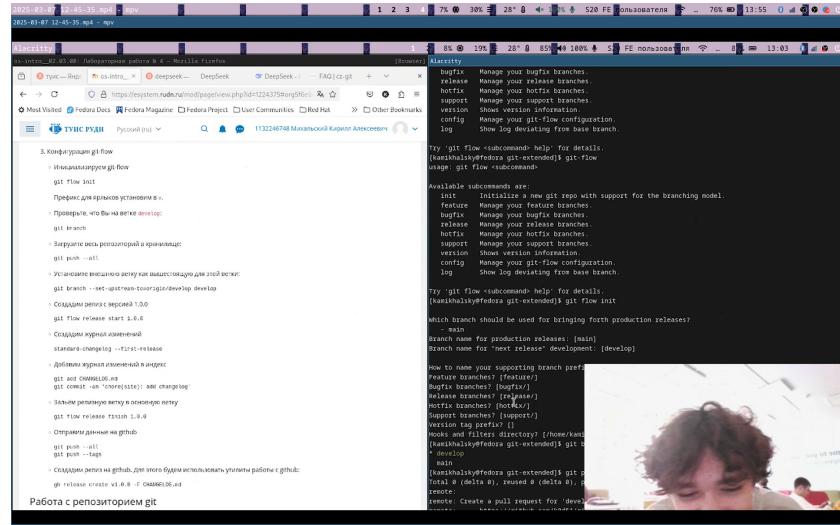


Рис. 4.4: установка4

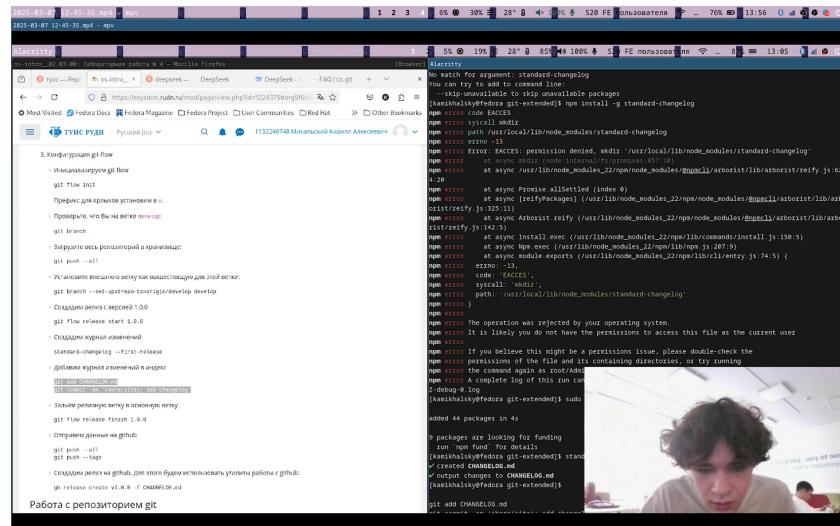


Рис. 4.5: установка5

1. Создал репозиторий git-extended. Сделал первый коммит и выложил на github. (рис. 4.6).

Рис. 4.6: коммит1

2. Инициализировал рпрт и изменил конфигурацию конфигов. (рис. 4.7).

Рис. 4.7: конфиг

Добавил новые файлы и отправил на github. (рис. 4.8).

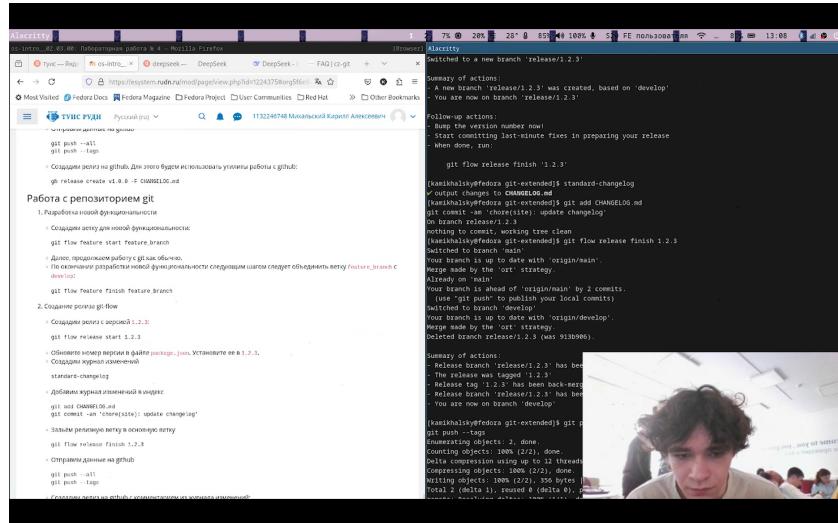


Рис. 4.8: коммит

3. Инициализировал git-flow (рис. ??).

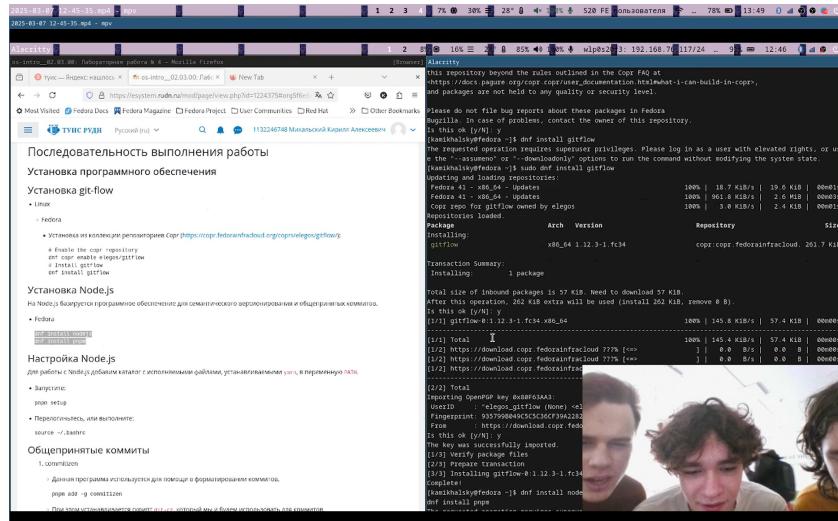


Рис. 4.9: инициализация

Проверил ветку, загрузил весь репозиторий. Установил внешнюю ветку как вышестоящую, создал релиз и журнал изменений. (рис. 4.10).

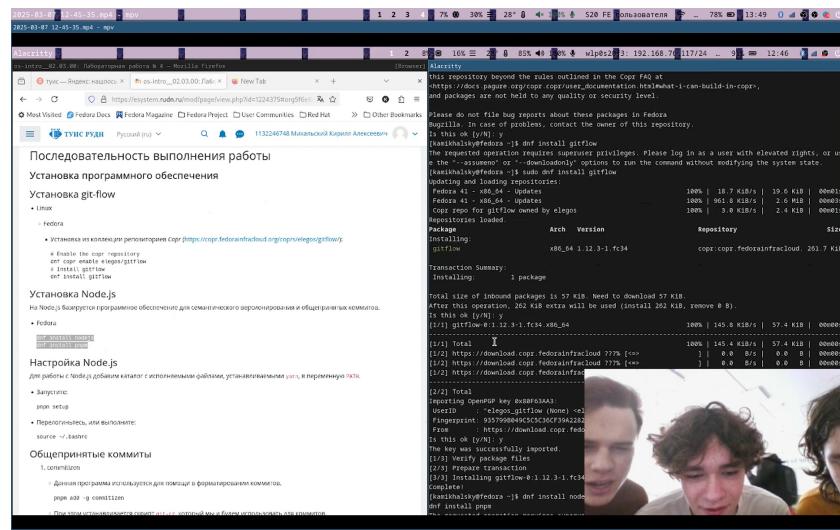


Рис. 4.10: проверка

Залил релизную ветку в основную и отправил на github. Создал релиз на github

1. Создал ветку для новой функциональности и объединил ветки. (рис. 4.11).

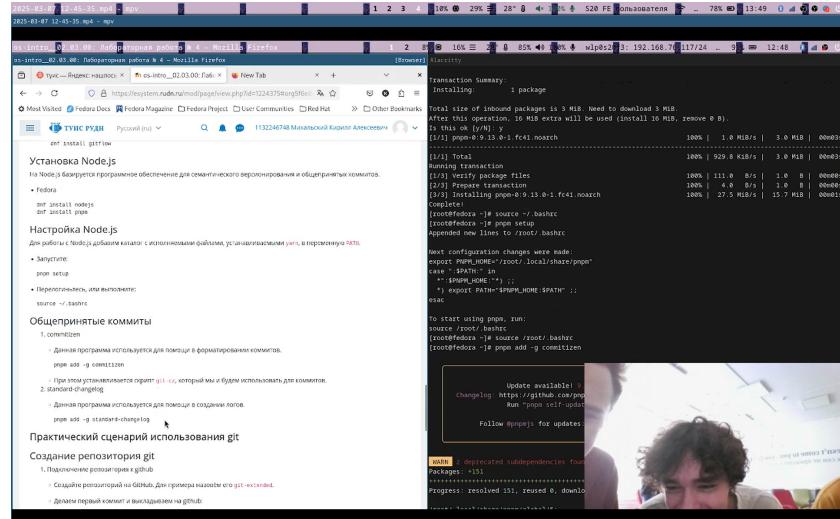


Рис. 4.11: процесс

2. Создал новый релиз, обновил версию. Создал журнал изменений и добавил его в индекс. (рис. 4.13).

```

2025-01-07 12:45:35 прб - prb
2025-01-07 12:45:35 прб - prb
root@kalinichuk-fedoraproject: ~ % cd /tmp/; git clone https://github.com/kalinichuk/fedora-git-extended.git
Cloning into 'fedora-git-extended'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done
Receiving objects: 100% (10/10), done
remote: Processing objects: 100% (10/10), done
remote: Writing objects: 100% (10/10), done
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: 
remote: To https://github.com/kalinichuk/fedora-git-extended.git
 * [new branch] master > origin/master
Branch 'master' set up to track remote branch 'master' from 'origin'.
warning: You appear to have cloned an empty repository.
root@kalinichuk-fedoraproject: ~ %

```

**Общепринятые коммиты**

- Дополнение
- Добавляется программа для помощи в форматировании коммитов.
- Установка
- Перегоняется, или вызывается:
- `git commit -m "commit"`

**Практический сценарий использования git**

**Создание репозитория git**

1. Подключение репозитория к github
- Создайте репозиторий на GitHub. Для примера назовите его `git-extended`.
- Добавьте первый коммит и выложить на github:

```

git commit -m "first commit"
git remote add origin https://github.com/kalinichuk/git-extended.git
git push -u origin master

```

2. Конфигурация обширных коммитов
- Конфигурация для пакета Node.js
- пакет
- Необходимо установить несколько параметров пакета.
- Использование пакета.
- Выложить пакет. Список лицензий для прм <https://nodejs.org/licenses/>. Предлагается выбирать лицензию от CC-BY-4.

Рис. 4.12: процесс

```

2025-01-07 12:45:35 прб - prb
2025-01-07 12:45:35 прб - prb
root@kalinichuk-fedoraproject: ~ % cd /tmp/; git clone https://github.com/kalinichuk/fedora-git-extended.git
Cloning into 'fedora-git-extended'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done
Receiving objects: 100% (10/10), done
remote: Processing objects: 100% (10/10), done
remote: Writing objects: 100% (10/10), done
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: 
remote: To https://github.com/kalinichuk/fedora-git-extended.git
 * [new branch] master > origin/master
Branch 'master' set up to track remote branch 'master' from 'origin'.
warning: You appear to have cloned an empty repository.
root@kalinichuk-fedoraproject: ~ %

```

**Практический сценарий использования git**

**Создание репозитория git**

1. Подключение репозитория к github
- Создайте репозиторий на GitHub. Для примера назовите его `git-extended`.
- Добавьте первый коммит и выложить на github:

```

git commit -m "first commit"
git remote add origin https://github.com/kalinichuk/git-extended.git
git push -u origin master

```

2. Конфигурация обширных коммитов
- Конфигурация для пакета Node.js
- пакет
- Необходимо установить несколько параметров пакета.
- Использование пакета.
- Выложить пакет. Список лицензий для прм <https://nodejs.org/licenses/>. Предлагается выбирать лицензию от CC-BY-4.

Рис. 4.13: процесс

Отправил релиз в github с комментариями из журнала. (рис. 4.14).

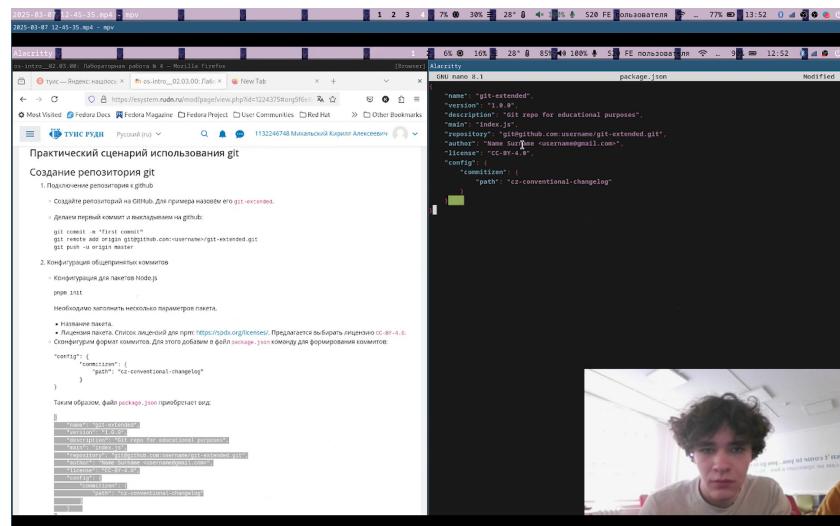


Рис. 4.14: завершение

## **5 Выводы**

Получил навыки правильной работы с репозиториями git.

## **Список литературы**