

# **Лабораторная работа №4**

**Отчет**

Арсакаев Дени

# 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>11</b>

## **Список иллюстраций**

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

## 2 Задание

Выполнить работу для тестового репозитория. Преобразовать рабочий репозиторий в репозитории с git-flow и обычными коммитами.

### 3 Выполнение лабораторной работы

Установка git-flow

```
[deniars@ARS ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для deniars:
https://copr.fedorainfracloud.org 100% | 652.0  B | 00m01s
Включение репозитория Copr. Обратите внимание, что этот
[deniars@ARS ~]$ sudo dnf install gitflow
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates ???% | 0.0  B | 00m01s
```

Установка Node.js

```
[deniars@ARS ~]$ sudo dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет      Арх.  Версия      Репоз      Размер
```

Устанавливаю pnpm

```
[deniars@ARS ~]$ sudo wget -qO- https://get.pnpm.io/install.sh | sh -
==> Downloading pnpm binaries 10.6.1
curl: (6) Could not resolve host: github.com
Install Error!
```

Запускаю pnpm

```
[deniars@ARS ~]$ pnpm setup
```

Выполняю

```
[deniars@ARS ~]$ source ~/.bashrc
[deniars@ARS ~]$ pnpm setup
```

Данная программа используется для помощи в форматировании коммитов.

```
[deniars@ARS ~]$ pnpm add -g commitizen
Progress: resolved 124, reused 0, downloaded 106, added
Progress: resolved 132, reused 0, downloaded 119, added
Progress: resolved 142, reused 0, downloaded 127, added
```

При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов.

Данная программа используется для помощи в создании логов.

```
[deniars@ARS ~]$ pnpm add -g standard-changelog
Progress: resolved 139, reused 138, downloaded 0, added
Progress: resolved 153, reused 150, downloaded 0, added
Progress: resolved 156, reused 150, downloaded 3, added
```

Создаю репозиторий с названием git-extended на GitHub

И скачиваю себе на Sway

```
[deniars@ARS ~]$ git clone --recursive git@github.com:DeniARS-2
28/git-extended.git
Клонирование в «git-extended»...
```

Делаю первый коммит и выкладываем на github:

```
[deniars@ARS git-extended]$ git commit -m "first commit"
Текущая ветка: main

Начальный коммит

[deniars@ARS git-extended]$ git remote add origin git@ggit@git
ub.com:DeniARS-228/git-extended.git
error: внешний репозиторий origin уже существует
```

Конфигурация для пакетов Node.js

Заполняю несколько параметров

```
[deniars@ARS git-extended]$ pnpm init
Wrote to /home/deniars/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": "",
  "license": "ISC",
  "packageManager": "pnpm@10.6.1"
}
[deniars@ARS git-extended]$
```

Получаю такой результат



```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:DeniARS-228/git-extended.git",
  "author": "Deni Arsakaev nomerabazyyy@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Добавляю новые файлы и выполняю коммит:

```
[deniars@ARS git-extended]$ git add .
[deniars@ARS git-extended]$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0
```

```
revert:  Reverts a previous commit
> feat:   A new feature
fix:     A bug fix
docs:    Documentation only changes
```

Инициализируем git-flow

```
[deniars@ARS git-extended]$ git flow init

Which branch should be used for bringing forth production releases?
- main
```

Префикс для ярлыков устанавливаю в v

Проверяю, что я на ветке develop:

```
[deniars@ARS git-extended]$ git branch
* develop
main
```

Загружаю весь репозиторий в хранилище:

```
[deniars@ARS 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:
```

Устанавливаю внешнюю ветку как вышестоящую для этой ветки и создаю релиз с версией 1.0.0

```
[deniars@ARS git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[deniars@ARS git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «release/1.0.0»
```

Отправим данные на github

```
[deniars@ARS git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.

[deniars@ARS git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
```

Создаю релиз на github. Для этого буду использовать утилиты работы с github:

```
[deniars@ARS git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
To get started with GitHub CLI, please run: gh auth login
Alternatively, populate the GH_TOKEN environment variable with a GitHub
```

Создаю релиз с версией 1.2.3:

```
[deniars@ARS git-extended]$ git flow release start 1.2.3
Переключились на новую ветку «release/1.2.3»
```

Обновляю номер версии в файле package.json и получаю такой результат

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repository for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:DeniARS-228/git-extended.git",
  "author": "Deni Arsakaev nomerabazyyy@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Добавляю журнал изменений в индекс

```
[deniars@ARS git-extended]$ git add CHANGELOG.md
[deniars@ARS git-extended]$ git commit -am 'chore(
```

Отправляю данные на github

```
[deniars@ARS git-extended]$ git push --all
[deniars@ARS git-extended]$ git push --tags
```

Создаю релиз на github с комментарием из журнала изменений с помощью команды `gh release create v1.2.3 -F CHANGELOG.md`

## 4 Выводы

Мы получили навыки работы с репозиториями git