

# **Лабораторная работа №4**

**Отчёт**

Александр Денисович Мосолов

# 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>15</b> |
|          | <b>Список литературы</b>              | <b>16</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Enable the copr repository . . . . .                                   | 7  |
| 3.2  | Install gitflow . . . . .  | 7  |
| 3.3  | Установка Node.js . . . . .  | 7  |
| 3.4  | Установка pnpm . . . . .   | 7  |
| 3.5  | Проверка . . . . .   | 8  |
| 3.6  | Клонируем репозиторий . . . . .  | 8  |
| 3.7  | Standard-changelog . . . . .   | 8  |
| 3.8  | Commitizen . . . . .   | 8  |
| 3.9  | Первый коммит . . . . .  | 8  |
| 3.10 | git push в ветку main . . . . .  | 9  |
| 3.11 | pnpm init . . . . .  | 9  |
| 3.12 | Изменяем package.json . . . . .  | 10 |
| 3.13 | Инициализируем git-flow . . . . .                                      | 10 |
| 3.14 | Ветка develop . . . . .  | 11 |
| 3.15 | git push –all . . . . .  | 11 |
| 3.16 | Внешняя ветка становится вышестоящей . . . . .                         | 11 |
| 3.17 | Создадим релиз с версией 1.0.0 . . . . .                               | 11 |
| 3.18 | Добавим журнал изменений в индекс . . . . .                            | 11 |
| 3.19 | Отправим данные на github . . . . .                                    | 12 |
| 3.20 | Создадим релиз на github . . . . .                                     | 12 |
| 3.21 | Создадим ветку для новой функциональности . . . . .                    | 12 |
| 3.22 | Создадим релиз с версией 1.2.3 . . . . .                               | 13 |
| 3.23 | Обновите номер версии в файле package.json . . . . .                   | 13 |
| 3.24 | Создадим журнал изменений . . . . .                                    | 13 |
| 3.25 | Добавим журнал изменений в индекс . . . . .                            | 13 |
| 3.26 | Зальём релизную ветку в основную ветку . . . . .                       | 14 |
| 3.27 | Отправим данные на github . . . . .                                    | 14 |
| 3.28 | Создадим релиз на github с комментарием из журнала изменений . . . . . | 14 |

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.[1]

## 2 Задание

Выполнить работу для тестового репозитория.

Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

### 3 Выполнение лабораторной работы

Установка из коллекции репозитория Copr (рис. 3.1):

```
[root@almos05 ~]# dnf copr enable elegos/gitflow
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.
```

Рис. 3.1: Enable the copr repository

Установка gitflow (рис. 3.2):

```
[root@almos05 ~]# dnf install gitflow
Copr repo for gitflow owned by elegos 2.4 kB/s |
```

Рис. 3.2: Install gitflow

Установка Node.js (рис. 3.3):

```
[root@almos05 ~]# dnf install nodejs
```

Рис. 3.3: Установка Node.js

Установка pnpm с помощью wget(рис. 3.4):

```
[almos05@almos05 ~]$ wget -qO- https://get.pnpm.io/install.sh | sh -
==> Downloading pnpm binaries 8.15.4
WARN using --force I sure hope you know what you are doing
Copying pnpm CLI from /tmp/tmp.5vqTY7X91H/pnpm to /home/almos05/.local/s
```

Рис. 3.4: Установка pnpm

Запускаем pnpm и выполняем (рис. 3.5):

```
[almos05@almos05 ~]$ source /home/almos05/.bashrc
```

Рис. 3.5: Проверка

Создаем пустой репозиторий и клонируем его (рис. 3.6):

```
[almos05@almos05 ~]$ git clone --recursive git@github.com:almos05/git-extended.git
Клонирование в «git-extended»...
```

Рис. 3.6: Клонирование репозитория

standard-changelog используется для помощи в создании логов (рис. 3.7):

```
[almos05@almos05 git-extended]$ pnpm add -g standard-changelog
Packages: +71
Progress: resolved 71, reused 0, downloaded 71, added 71, done

/home/almos05/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0
```

Рис. 3.7: Standard-changelog

Commitizen используется для помощи в форматировании коммитов (рис. 3.8):

```
[almos05@almos05 git-extended]$ pnpm add -g commitizen
Packages: +138
Downloading registry.npmjs.org/typescript/5.4.2: 5.82 MB/5.82 MB, done
Progress: resolved 289, reused 71, downloaded 138, added 138, done

/home/almos05/.local/share/pnpm/global/5:
```

Рис. 3.8: Commitizen

Создаем пустой файл README.md, фиксируем изменения и делаем первый коммит (рис. 3.9):

```
[almos05@almos05 git-extended]$ touch README.md
[almos05@almos05 git-extended]$ git add .
[almos05@almos05 git-extended]$ git commit -m "first commit"
[main (корневой коммит) 993240c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
```

Рис. 3.9: Первый коммит



Делаем git push в ветку main (рис. 3.10):

```
[almos05@almos05 git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 874 байта | 874.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:almos05/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.10: git push в ветку main

Конфигурация для пакетов Node.js (рис. 3.11):

```
[almos05@almos05 ~]$ pnpm init
Wrote to /home/almos05/package.json

{
  "name": "almos05",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
[almos05@almos05 ~]$
```

Рис. 3.11: pnpm init

Изменяем package.json (рис. 3.12):

```
GNU nano 7.2 /home/almos05/package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:almos05/git-extended.git",
  "author": "Aleksandr Mosolov tenderboylove@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.12: Изменяем package.json

Добавим новые файлы:

git add .

Выполним коммит:

git cz

Отправим на github:

git push

Инициализируем git-flow, префикс для ярлыков установим в v (рис. 3.13):

```
[almos05@almos05 git-extended]$ git flow init -f

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

Which branch should be used for integration of the "next release"?
  - develop
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]
```

Рис. 3.13: Инициализируем git-flow

Проверьте, что Вы на ветке develop (рис. 3.14):

```
[almos05@almos05 git-extended]$ git branch
* develop
main
```

Рис. 3.14: Ветка develop

Загружаем весь репозиторий в хранилище (рис. 3.15):

```
[almos05@almos05 git-extended]$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
```

Рис. 3.15: git push --all

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 3.16):

```
[almos05@almos05 git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[almos05@almos05 git-extended]$
```

Рис. 3.16: Внешняя ветка становится вышестоящей

Создадим релиз с версией 1.0.0 (рис. 3.17):

```
[almos05@almos05 git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «release/1.0.0»
```

Рис. 3.17: Создадим релиз с версией 1.0.0

Создадим журнал изменений (рис. 3.18):

standard-changelog --first-release

И добавим журнал изменений в индекс

```
[almos05@almos05 git-extended]$ git add CHANGELOG.md
[almos05@almos05 git-extended]$ git commit -m 'chore(site): add changelog'
[release/1.0.0 1ec0408] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.18: Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

git flow release finish 1.0.0

Отправим данные на github (рис. 3.19):

```
[alnos05@alnos05 git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При скатии изменений используется до 4 потоков
Скатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.80 Киб | 2.80 Миб/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:alnos05/git-extended.git
   9028aa5..6400471 develop -> develop
   9028aa5..00220f1 main -> main
[alnos05@alnos05 git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 165 байтов | 165.00 Киб/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:alnos05/git-extended.git
   * [new tag]         v1.0.0 -> v1.0.0
[alnos05@alnos05 git-extended]$
```

Рис. 3.19: Отправим данные на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 3.20):

```
[alnos05@alnos05 git-extended]$ gh release create v1.0.0 -F CHANGELOG.r
https://github.com/alnos05/git-extended/releases/tag/v1.0.0
[alnos05@alnos05 git-extended]$
```

Рис. 3.20: Создадим релиз на github

Создадим ветку для новой функциональности (рис. 3.21):

```
[alnos05@alnos05 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:
```

Рис. 3.21: Создадим ветку для новой функциональности

Создадим релиз с версией 1.2.3 (рис. 3.22):

```
[almos05@almos05 git-extended]$ git flow release start 1.2.3
Переключились на новую ветку «release/1.2.3»

Summary of actions:
- A new branch 'release/1.2.3' was created, based on 'develop'
- You are now on branch 'release/1.2.3'
```

Рис. 3.22: Создадим релиз с версией 1.2.3

Обновите номер версии в файле package.json. Установите её в 1.2.3 (рис. 3.23):

```
GNU nano 7.2 package.json Изменён

{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:almos05/git-extended.git",
  "author": "Aleksandr Mosolov tenderboy@live.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.23: Обновите номер версии в файле package.json

Создадим журнал изменений (рис. 3.24):

```
[almos05@almos05 git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[almos05@almos05 git-extended]$ git add CHANGELOG.md
```

Рис. 3.24: Создадим журнал изменений

Добавим журнал изменений в индекс

git add CHANGELOG.md, сделаем новый коммит(рис. 3.25):

```
[almos05@almos05 git-extended]$ git commit -m 'chore(site): update changelog'
[feature/feature_branch 5212aac] chore(site): update changelog
1 file changed, 12 insertions(+)
[almos05@almos05 git-extended]$
```

Рис. 3.25: Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку (рис. 3.26):

```
[almos05@almos05 git-extended]$ git flow release finish 1.2.3
```

Рис. 3.26: Зальём релизную ветку в основную ветку

Отправим данные на github (рис. 3.27):

```
[almos05@almos05 git-extended]$ git push --all
gПеречисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.82 КиБ | 2.82 МБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
git To github.com:almos05/git-extended.git
 6400471..e34069d develop -> develop
 00220f1..a8c6d30 main -> main
* [new branch]      feature/feature_branch -> feature/feature_branch
[almos05@almos05 git-extended]$ git push --tags
```

Рис. 3.27: Отправим данные на github

Зальём релизную ветку в основную Создадим релиз на github с комментарием из журнала изменений]):

```
[almos05@almos05 git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/almos05/git-extended/releases/tag/v1.2.3
[almos05@almos05 git-extended]$
```

Рис. 3.28: Создадим релиз на github с комментарием из журнала изменений

## 4 Выводы

В ходе работы мы получили навыки правильной работы с репозиториями git.

## Список литературы

1. Kulyabov. Архитектура компьютеров и операционные системы. Раздел "Операционные системы" (09.03.03, НПИбд).