

Отчет по лабораторной работе №4

Операционные системы

Гашимова Э.Э.

Российский университет дружбы народов, Москва, Россия

Цель работы

Получение навыков правильной работы с репозиториями git.

Задание


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

Теоретическое введение

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета `git-flow`. Общая информация Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде. 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 сливается с веткой `develop`. Когда работа над веткой релиза `release` завершена, она сливается в ветки `develop` и `master`. Если в `master` обнаружена проблема, из `master` создаётся ветка `hotfix`. Когда работа над веткой исправления `hotfix` завершена, она сливается в ветки `develop` и `master`.

Выполнение лабораторной работы

1. Открываем терминал и входим в режим суперпользователя, устанавливаем gitflow.

A terminal window with a dark background and light-colored text. The command 'dnf copr enable elegos/gitflow' is entered and executed. The output shows the command being processed and the package being installed.

```
dnf copr enable elegos/gitflow
```

Рис. 1: Установка gitflow в режиме суперпользователя

2. Устанавливаем npm и nodejs.

A screenshot of a terminal window with a dark background. The text 'dnf install npm' is displayed in a light-colored, monospaced font. The 'dnf' command is used for package management in Fedora Linux, and 'install npm' indicates the installation of Node Package Manager (npm), which is bundled with Node.js.

```
dnf install npm
```

Рис. 2: Установка приложений

3. Настраиваем nodejs.



Рис. 3: Настройка node.js

4. Настраиваем commitizen и standard-changelog.

A screenshot of a terminal window with a dark background. The text 'pnpm add -g commitizen' is displayed in a light-colored, monospaced font, representing a command to install the commitizen package globally using pnpm.

```
pnpm add -g commitizen
```

Рис. 4: Настройка программ

5. Создаем репозиторий git, настраиваем его и делаем в него первый коммит.

```
git clone --recursive
```

Рис. 5: Создание репозитория, первый коммит

6. Настраиваем пакет файлов nodejs. В файле `package.json` меняем необходимые данные.

```
git push -u
```

Рис. 6: Настройка пакета



Рис. 7: Изменения файла

7. Выполняем коммит. Выкладываем на github.



Рис. 8: Выполнение коммита

```
$ git push --all
```

Рис. 9: Команда push

8. Инициализируем gitflow, проверяем, на какой ветке мы находимся в данный момент, после чего загружаем весь репозиторий в хранилище.

A screenshot of a terminal window with a black background and white text. The text shows the command 'git flow release start 1.0.0' being entered. The text is slightly blurred, suggesting it might be a video frame or a low-resolution screenshot.

```
git flow release start 1.0.0
```

Рис. 10: Инициализация gitflow

9. Устанавливаем внешнюю ветку как вышестоящую для этой ветки. Создаем релиз с версией 1.0.0 и журнал изменений..

```
git add CHANGELOG.md'
```

Рис. 11: Установка внешней ветки

```
$ git flow release finish 1.0.0
```

Рис. 12: Создание релиза и журнала изменений

10. Заливаем релизную ветку в основную, добавляем журнал изменений в индекс, после чего заливаем релизную ветку в основную.

```
git push --all
```

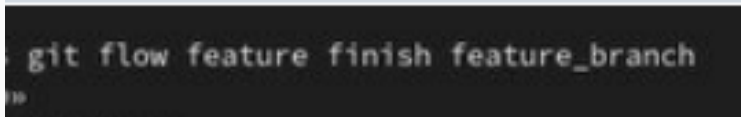
Рис. 13: Добавление релизной ветки в основную

11. Отправляем данные и теги на гитхаб.

```
$ gh release create v1.0.0 -F CHANGELOG.md
```

Рис. 14: Команды push -all и push -tags

12. Создаем релиз на гитхабе. Создаем ветку для новой функциональности.

A screenshot of a terminal window with a dark background. The text 'git flow feature finish feature_branch' is displayed in a light-colored, monospaced font. The text is slightly blurred, suggesting it might be a video frame or a screenshot taken quickly. The command is split across two lines: 'git flow feature finish' on the first line and 'feature_branch' on the second line.

```
git flow feature finish feature_branch
```

Рис. 15: Создание новой ветки

13. Объединяем новую ветку с develop. Создаём релиз с версией 1.2.3.
14. Изменяем номер версии в файле package.json.
15. Заливаем релизную ветку в основную. Отправляем данные на гитхаб.
16. Создаём релиз на гитхабе с комментарием из журнала изменений.

В процессе выполнения лабораторной работы я приобрела навыки правильной работы с репозиториями git.