

Лабораторная работа №4

Презентация

Андрюшин Н. С.

25 февраля 2024

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Андрюшин Никита Сергеевич
- Студент
- Российский университет дружбы народов
- 1132231848@pfur.ru

Цель

Получение навыков правильной работы с репозиториями git

Задачи

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

Установка ПО

Для начала подключим репозиторий, из которого можно скачать gitflow

```
[nsandryushin@nsandryushin ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для nsandryushin:
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне
.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого р
епозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: у
Репозиторий успешно подключен.
```

Рис. 1: Подключение репозитория

Установка ПО

После этого установим сам gitflow

```
[nsandryushin@nsandryushin ~]$ dnf install gitflow
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на
большинстве систем - под именем пользователя root).
[nsandryushin@nsandryushin ~]$ sudo dnf install gitflow
Скоп 0ero for gitflow owned by elegos   1.5 kB/s | 1.5 kB    00:01
Fedora 39 - x86_64 - Updates           50 kB/s | 15 kB    00:00
Fedora 39 - x86_64 - Updates           622 kB/s | 2.4 MB   00:03
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, В
с 03 мар 2024 13:51:06.
Зависимости разрешены.
=====
Пакет      Архитектура      Версия      Репозиторий      Размер
=====
Установка:
gitflow x86_64 1.12.3-1.fc34
          copr:fedorainfracloud.org:elegos:gitflow 57 k

Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]: у
```

Рис. 2: Установка gitflow

Установка ПО

Теперь установим NodeJs

```
[nsandryushin@nsandryushin ~]$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:25 назад, В
с 03 мар 2024 13:51:06.
Зависимости разрешены.

=====
Пакет          Архитектура      Версия      Репозиторий      Размер
=====
Установка:
nodejs          x86_64        1:20.10.0-3.fc39    updates   48 k
Установка зависимостей:
nodejs-libs     x86_64        1:20.10.0-3.fc39    updates   15 M
Установка слабых зависимостей:
nodejs-docs     noarch       1:20.10.0-3.fc39    updates   8.1 M
nodejs-full-libs x86_64        1:20.10.0-3.fc39    updates   8.5 M
nodejs-npm      x86_64        1:10.2.3-1.20.10.0-3.fc39    updates   2.2 M
=====
Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 34 М
Объем изменений: 183 М
Продолжить? [д/н]: у
```

Рис. 3: Установка NodeJs

Установка ПО

Установим pnpm

```
[nsandryushin@nsandryushin ~]$ sudo 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.rUuc5xSbho/pnpm to /home/nsandryushin/.local/share/pnpm/pnpm
Appended new lines to /home/nsandryushin/.bashrc

Next configuration changes were made:
export PNPM_HOME="/home/nsandryushin/.local/share/pnpm"
case ":$PATH:" in
  *":$PNPM_HOME:*") ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/nsandryushin/.bashrc
```

Рис. 4: Установка pnpm

Установка ПО

Запустим pnpm

```
[nsandryushin@nsandryushin ~]$pnpm setup  
No changes to the environment were made. Everything is already up to date.  
[nsandryushin@nsandryushin ~]$source ~/.bashrc
```

Рис. 5: Запуск pnpm

Установка ПО

И установим с помощью него Commitizen

```
[nsandryushin@nsandryushin ~]$ pnpm add -g commitizen
Packages: +152
+-----+
Downloading registry.npmjs.org/typescript/5.3.3: 5.76 MB/5.76 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/home/nsandryushin/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 15.7s
[nsandryushin@nsandryushin ~]$ pnpm add -g standard-changelog
Packages: +56
+-----+
Progress: resolved 208, reused 152, downloaded 56, added 56, done

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

Done in 6.9s
```

Рис. 6: Установка Commitizen

Настройка репозитория

Создадим тестовый репозиторий git-extended

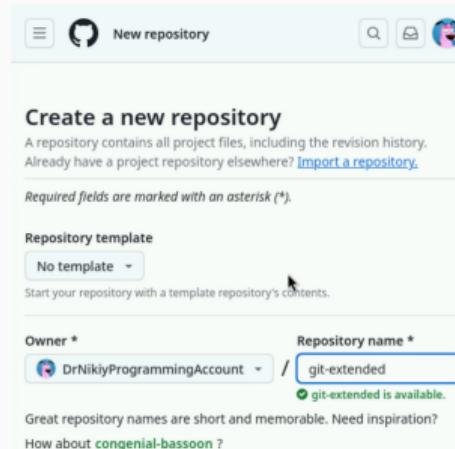


Рис. 7: Создание тестового репозитория

Настройка репозитория

И клонируем его себе на компьютер

```
[nsandryushin@nsandryushin ~]$ git clone --recursive git@github.com:DrNikiyProgrammingAccount/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
```

Рис. 8: Клонирование репозитория

Настройка репозитория

Создадим какой-нибудь файл и проиндексируем его с помощью git add

```
[nsandryushin@nsandryushin git-extended]$ touch README.md  
[nsandryushin@nsandryushin git-extended]$ git add .
```

Рис. 9: Создание файла в репозитории и индексирование

Настройка репозитория

Теперь сделаем соответствующий коммит

```
[nsandryushin@nsandryushin git-extended]$ git commit -m "first commit"
[main (корневой коммит) ea3a7de] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Рис. 10: Создание коммита

Настройка репозитория

И добавим ветку

```
[nsandryushin@nsandryushin git-extended]$ git remote add origin git@gith  
ub.com:DrNikiyProgrammingAccount/git-extended.git
```

Рис. 11: Добавление ветки

Настройка репозитория

Теперь запушим её обратно на гитхаб

```
[nsandryushin@nsandryushin git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 882 байта | 882.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:DrNikiyProgrammingAccount/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[nsandryushin@nsandryushin git-extended]$ []
```

Рис. 12: Загрузка изменений на ГитХаб

Продвинутое создание коммитов

Теперь проинициализируем pnpm

```
[nsandryushin@nsandryushin git-extended]$ pnpm init
Wrote to /home/nsandryushin/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"
}
```

Рис. 13: Инициализация pnpm

Продвинутое создание коммитов

После инициализации создаётся файл package.json, который нужно изменить следующим образом

```
GNU nano 7.2          package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "git@github.com:DrNikiyProgrammingAccount/git-extended.>
  "author": "Nikita Andryushin mega_nikitos111@mail.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 14: Редактирование файла package.json

Продвинутое создание коммитов

Сделаем коммит, но уже с помощью CZ

```
[nsandryushin@nsandryushin git-extended]$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? 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) readme.md
? Write a short, imperative tense description of the change (max 83
  chars):
  (10) added file
? 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
[main 4c23d75] feat(readme.md): added file
  1 file changed, 15 insertions(+)
  create mode 100644 package.json
```

Рис. 15: Создание коммита с помощью CZ

Продвинутое создание коммитов

Загрузим изменения на гитхаб

```
[nsandryushin@nsandryushin git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.19 КиБ | 1.19 МиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:DrNikiyProgrammingAccount/git-extended.git
  ea3a7de..4c23d75  main -> main
[nsandryushin@nsandryushin git-extended]$ git flow init
Fatal: Working tree contains unstaged changes. Aborting.
```

Рис. 16: Загрузка изменений на GitHub

Использование gitflow

Теперь проинициализируем gitflow. Укажем названия веток и префикс для версий

```
[nsandryushin@nsandryushin 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
Hooks and filters directory? [/home/nsandryushin/git-extended/.git/hooks]
```

Рис. 17: Инициализация GitFlow

Использование gitflow

Выведем список веток и убедимся, что мы находимся в develop, и запушим изменения на сервер

```
[nsandryushin@nsandryushin git-extended]$ git branch
* develop
  main
[nsandryushin@nsandryushin git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/DrNikiyProgrammingAccount/git-extended/p
ull/new/develop
remote:
remote:
To github.com:DrNikiyProgrammingAccount/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 18: Просмотр веток и загрузка изменений

Использование gitflow

Переключимся на ветку develop, после чего создадим ветку релиза, где создадим changelog

```
[nsandryushin@nsandryushin git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[nsandryushin@nsandryushin git-extended]$ git flow release start 1.0.0
Переключились 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'

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

Рис. 19: Смена ветки и создание ветки релиза, и создание Changelog

Использование gitflow

Проиндексируем changelog и сделаем коммит

```
[nsandryushin@nsandryushin git-extended]$ git add CHANGELOG.md
[nsandryushin@nsandryushin git-extended]$ git commit -am 'chore(site): a
dd changelog'
[release/1.0.0 74cc481] chore(site): add changelog
 1 file changed, 9 insertions(+)
 create mode 100644 CHANGELOG.md
```

Рис. 20: Создание коммита с changelog

Использование gitflow

Теперь сольём ветку release с веткой develop

```
[nsandryushin@nsandryushin git-extended]$ git flow release finish 1.0.0
Branches 'main' and 'origin/main' have diverged.
And local branch 'main' is ahead of 'origin/main'.
Уже на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++-
 1 file changed, 9 insertions(+)
 create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 74cc481).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'
```

Рис. 21: Слияние веток

Использование gitflow

Загрузим изменения в гитхаб

```
[nsandryushin@nsandryushin git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.83 КиБ | 2.83 МиБ/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно и
использовано пакетов 0
To github.com:DrNikiyProgrammingAccount/git-extended.git
  4c23d75..cb6db1f develop -> develop
  4c23d75..2697000 main -> main
[nsandryushin@nsandryushin 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.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
использовано пакетов 0
To github.com:DrNikiyProgrammingAccount/git-extended.git
 * [new tag]      v1.0.0 -> v1.0.0
```

Рис. 22: Загрузка изменений на GitHub

Использование gitflow

Создадим релиз из changelog'a

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

Рис. 23: Создание релиза

Работа с релизами

Создадим ветку feature и сразу сольём её с develop

```
[nsandryushin@nsandryushin 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

[nsandryushin@nsandryushin git-extended]$ ls
CHANGELOG.md package.json README.md
[nsandryushin@nsandryushin 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 сбрана).

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'
```

Рис. 24: Создание ветки feature и её слияние

Работа с релизами

Создадим ветку релиза

```
[nsandryushin@nsandryushin 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'

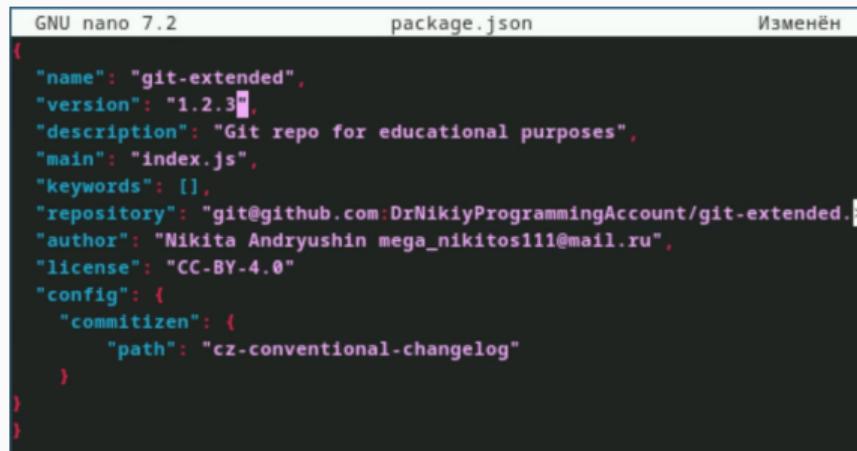
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'
```

Рис. 25: Создание ветки релиза

Работа с релизами

И в package.json сменим версию



```
GNU nano 7.2          package.json          Изменён
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "git@github.com:DrNikiyProgrammingAccount/git-extended.",
  "author": "Nikita Andryushin mega_nikitos11@mail.ru",
  "license": "CC-BY-4.0"
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 26: Изменение версии

Работа с релизами

Теперь создадим журнал изменений, проиндексируем его и сольём ветку с ним в ветку develop

```
[nsandryushin@nsandryushin git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[nsandryushin@nsandryushin git-extended]$ git add CHANGELOG.md
[nsandryushin@nsandryushin git-extended]$ git commit -am 'chore(site): update changelog'
[rebaseline/1.2.3 f6dd6c7] chore(site): update changelog
 2 files changed, 5 insertions(+), 1 deletion(-)
[nsandryushin@nsandryushin git-extended]$ git flow release finish 1.2.3
Переключаемся на ветку `main`
Эта ветка соответствует `origin/main`.
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)
Переключаемся на ветку `develop`
Эта ветка соответствует `origin/develop`.
Merge made by the 'ort' strategy.
 CHANGELOG.md | 4 +---
 package.json | 2 ++
 2 files changed, 5 insertions(+), 1 deletion(-)
Бранч release/1.2.3 удален (бранч f6dd6c7).

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'
```

Рис. 27: Создание журнала изменений, его индексация и объединение с основной веткой

Работа с релизами

Загрузим изменения в гитхаб и создадим релиз

```
[nsandryushin@nsandryushin git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.79 КиБ | 475.00 КиБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:DrNikiyProgrammingAccount/git-extended.git
  cbedbf..5ffa4cc develop -> develop
  2697000..77d8ac4 main -> main
[nsandryushin@nsandryushin git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 167 байтов | 167.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:DrNikiyProgrammingAccount/git-extended.git
 * [new tag]      v1.2.3 -> v1.2.3
[nsandryushin@nsandryushin git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/DrNikiyProgrammingAccount/git-extended/releases/tag/v1.2.3
```

Рис. 28: Загрузка изменений в гитхаб и создание релиза

Выводы

В результате выполнения лабораторной работы были получены навыки работы с расширенными возможностями git, а также были созданы релизы к репозиторию и дополнительные ветки, которые автор научился сливать воедино