

Лабораторная работа №4

Дисциплина: Архитектура компьютера

Орлов Илья Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	12
	Список литературы	13

Список иллюстраций

4.1	Установка необходимого ПО	8
4.2	Установка через npm	8
4.3	Создание репозитория	8
4.4	Конфигурация package.json	9
4.5	Отправление коммита на GitHub	9
4.6	Использование git-flow	10
4.7	Загрузка релиза на github	11
4.8	Работа с новой веткой и загрузка релиза	11

Список таблиц

1 Цель работы

Получение навыков продвинутой работы с репозиториями git и релизами.

2 Задание

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

3 Теоретическое введение

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.

Семантическое версионирование описывается в манифесте семантического версионирования. Кратко его можно описать следующим образом: Версия задаётся в виде кортежа МАЖОРНАЯ_ВЕРСИЯ.МИНОРНАЯ_ВЕРСИЯ.ПАТЧ. Номер версии следует увеличивать: МАЖОРНУЮ версию, когда сделаны обратно несовместимые изменения API. МИНОРНУЮ версию, когда вы добавляете новую функциональность, не нарушая обратной совместимости. ПАТЧ-версию, когда вы делаете обратно совместимые исправления. Дополнительные обозначения для предрелизных и билд-метаданных возможны как дополнения к МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ формату.

Спецификация Conventional Commits: Соглашение о том, как нужно писать сообщения commit'ов. Совместимо с SemVer. Даже вернее сказать, сильно связано с семантическим версионированием. Регламентирует структуру и основные типы коммитов.

4 Выполнение лабораторной работы

Устанавливаю nodejs, пакетный менеджер для него pnpm и gitflow. (рис. 4.1)

```
isorlov@isorlov:~$ sudo dnf install nodejs pnpm
[sudo] пароль для isorlov:
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "nodejs-1:22.18.0-1.fc42.x86_64" уже установлен.
Пакет "pnpm-10.9.0-1.fc42.noarch" уже установлен.

Нечего делать.
isorlov@isorlov:~$ git clone https://github.com/petervanderdoes/gitflow-avh.git
cd gitflow-avh
```

Рис. 4.1: Установка необходимого ПО

Устанавливаю через pnpm commitizen и standard-changelog. (рис. 4.2)

```
isorlov@isorlov:~$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 185, reused 151, downloaded 34, added 0, done
Done in 8.7s using pnpm v10.9.0
```

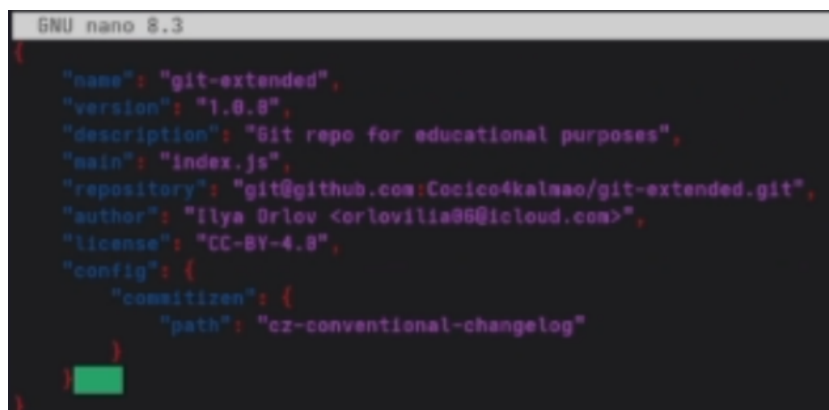
Рис. 4.2: Установка через pnpm

Создаю новый репозиторий и делаю там первый коммит. (рис. 4.3)

```
isorlov@isorlov:~$ gh repo create --public git-extended
✓ Created repository Cocico4kalmao/git-extended on github.com
https://github.com/Cocico4kalmao/git-extended
isorlov@isorlov:~$ git clone --recursive git@github.com:Cocico4kalmao/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
isorlov@isorlov:~$ cd git-extended/
bash: cd: git-extended/: Нет такого файла или каталога
isorlov@isorlov:~$ cd git-extended/
bash: cd: git-extended/: Нет такого файла или каталога
isorlov@isorlov:~$ cd git-extended
isorlov@isorlov:~/git-extended$ git add .
isorlov@isorlov:~/git-extended$ touch README.md echo text > README.md
isorlov@isorlov:~/git-extended$ git push
```

Рис. 4.3: Создание репозитория

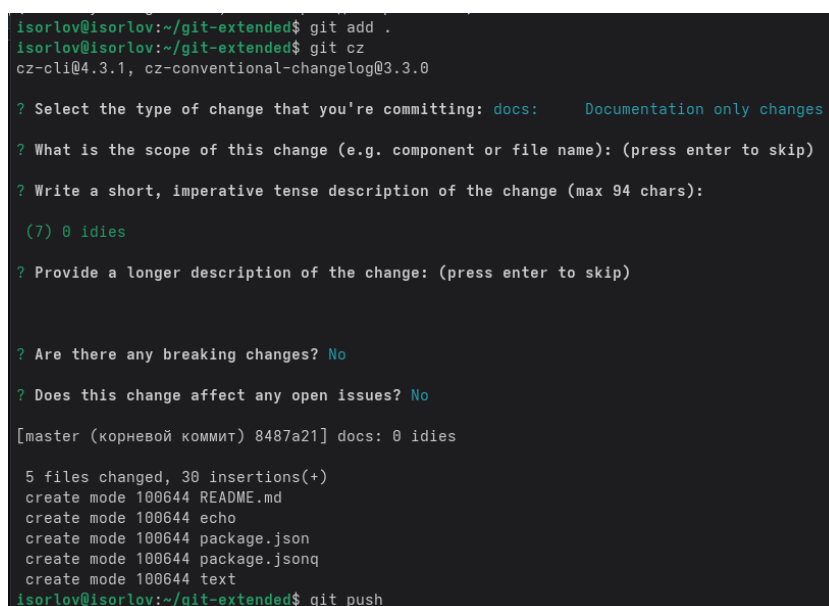
Инициализирую и конфигурирую общепринятые коммиты в созданной директории через редактирование package.json. (рис. 4.4)



```
GNU nano 8.3
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Cocico4kalmao/git-extended.git",
  "author": "Ilya Orlov <orlovilia06@icloud.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.4: Конфигурация package.json

Добавляю в индекс изменения, создаю коммит и отправляю на удаленный репозиторий. (рис. 4.5)



```
isorlov@isorlov:~/git-extended$ git add .
isorlov@isorlov:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: docs: Documentation only changes
? What is the scope of this change (e.g. component or file name): (press enter to skip)
? Write a short, imperative tense description of the change (max 94 chars):
(7) 0 idies
? 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

[master (корневой коммит) 8487a21] docs: 0 idies

5 files changed, 30 insertions(+)
create mode 100644 README.md
create mode 100644 echo
create mode 100644 package.json
create mode 100644 package.jsonq
create mode 100644 text
isorlov@isorlov:~/git-extended$ git push
```

Рис. 4.5: Отправление коммита на GitHub

Инициализирую в репозитории git flow и создаю 1 релиз в только что созданной ветке develop. (рис. 4.6)

```

isorlov@isorlov:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
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/isorlov/git-extended/.git/hooks]
isorlov@isorlov:~/git-extended$ git branch
* develop
  master
isorlov@isorlov:~/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:
remote:   https://github.com/Cocico4kalmao/git-extended/pull/new/develop
remote:
To github.com:Cocico4kalmao/git-extended.git
* [new branch]    develop -> develop
isorlov@isorlov:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
isorlov@isorlov:~/git-extended$ git flow release start 1.0.0

```

Рис. 4.6: Использование git-flow

Создаю список изменений через standard changelog, заканчиваю релиз и вы-
гружаю на удаленный репозиторий изменения. (рис. 4.7)

```

1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
isorlov@isorlov:~/git-extended$ git flow release finish 1.0.0
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
Ваша ветка опережает «origin/master» 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 | 4 ++++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 92041d1).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- 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'

isorlov@isorlov:~/git-extended$ git push --all

```

Рис. 4.7: Загрузка релиза на github

Инициализирую ветку feature для работы над новой функциональностью, готовлю релиз и загружаю на github. (рис. 4.8)

```

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'

isorlov@isorlov:~/git-extended$ nano package.json
isorlov@isorlov:~/git-extended$ nano package.json
isorlov@isorlov:~/git-extended$ standart-changelog
bash: standart-changelog: команда не найдена...
isorlov@isorlov:~/git-extended$ standard-changelog
isorlov@isorlov:~/git-extended$ git add .
isorlov@isorlov:~/git-extended$ git status
Текущая ветка: release/1.2.3
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   CHANGELOG.md
    изменено:   package.json

isorlov@isorlov:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: chore:   Other changes that don't modify src or test files
? What is the scope of this change (e.g. component or file name): (press enter to skip) site
? Write a short, imperative tense description of the change (max 87 chars):
(13) add changelog
? 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
[release/1.2.3 0e96d48] chore(site): add changelog
2 files changed, 5 insertions(+), 1 deletion(-)
isorlov@isorlov:~/git-extended$ git flow release finish 1.2.3

```

Рис. 4.8: Работа с новой веткой и загрузка релиза

5 Выводы

В ходе выполнения лабораторной работы я получил навыки правильной работы с репозиториями git.

Список литературы