

Лабораторная работа №4

дисциплина: Архитектура компьютера

Мазурский Александр Дмитриевич

Содержание

| | |
|---|-----------|
| 1 Цель работы | 5 |
| 2 Задание | 6 |
| 3 Теоретическое введение | 7 |
| 4 Выполнение лабораторной работы | 8 |
| 5 Выводы | 13 |
| Список литературы | 14 |

Список иллюстраций

| | |
|---|----|
| 4.1 Установка необходимого ПО | 8 |
| 4.2 Установка через pnpm | 9 |
| 4.3 Создание репозитория | 9 |
| 4.4 Конфигурация package.json | 10 |
| 4.5 Отправление коммита на GitHub | 10 |
| 4.6 Использование git-flow | 11 |
| 4.7 Загрузка релиза на github | 11 |
| 4.8 Работа с новой веткой и загрузка релиза | 12 |

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 Дриссеном. 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)

```
pnpm
Installing dependencies:
nodejs-libs                               noarch    9.13.0-1.fc41
Installing weak dependencies:
nodejs-docs                                 x86_64   1:22.14.0-2.fc41
nodejs-full-i18n                            x86_64   1:22.14.0-2.fc41
nodejs-npm                                 x86_64   1:10.9.2-1.22.14.0.2.fc41

Transaction Summary:
Installing:          6 packages

Total size of inbound packages is 42 MiB. Need to download 42 MiB.
After this operation, 225 MiB extra will be used (install 225 MiB, remove 0 B).
Is this ok [y/N]: y
[1/6] nodejs-1:22.14.0-2.fc41.x86_64
[2/6] pnpm-0:9.13.0-1.fc41.noarch
[3/6] nodejs-npm-1:10.9.2-1.22.14.0.2.fc41.x86_64
[4/6] nodejs-docs-1:22.14.0-2.fc41.noarch
[5/6] nodejs-full-i18n-1:22.14.0-2.fc41.x86_64
[6/6] nodejs-libs-1:22.14.0-2.fc41.x86_64
-----
[6/6] Total
Running transaction
[1/8] Verify package files
[2/8] Prepare transaction
[3/8] Installing nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/8] Installing nodejs-1:22.14.0-2.fc41.x86_64
[5/8] Installing pnpm-0:9.13.0-1.fc41.noarch
[6/8] Installing nodejs-npm-1:10.9.2-1.22.14.0.2.fc41.x86_64
[7/8] Installing nodejs-full-i18n-1:22.14.0-2.fc41.x86_64
[8/8] Installing nodejs-docs-1:22.14.0-2.fc41.noarch
Complete!
[admazurskiy@admazurskiy ~]$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/develop/contrib/install-nodejs-and-pnpm.sh | sh
```

Рис. 4.1: Установка необходимого ПО

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

```

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
+-----+
Progress: resolved 151, reused 0, downloaded 151, added 151, done

/home/admazurskiy/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 3.4s
[admazurskiy@admazurskiy ~]$ pnpm add -g standard-changelog
ERR_PNPM_FETCH_404 GET https://registry.npmjs.org/standard-changelog: Not Found - 404

This error happened while installing a direct dependency of /home/admazurskiy/.local/share/pnpm/global/5
standard-changelog is not in the npm registry, or you have no permission to fetch it.

No authorization header was set for the request.
Progress: resolved 1, reused 1, downloaded 0, added 0
[admazurskiy@admazurskiy ~]$ pnpm add -g standard-changelog
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +39
+-----+
Progress: resolved 190, reused 151, downloaded 39, added 39, done

/home/admazurskiy/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

```

Рис. 4.2: Установка через pnpm

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

```

[admazurskiy@admazurskiy ~]$ gh repo create --public git-extended
✓ Created repository nowherewashere/git-extended on GitHub
https://github.com/nowherewashere/git-extended
[admazurskiy@admazurskiy ~]$ git clone --recursive git@github.com:nowherewashere/git-extended.git
Cloning into 'git-extended'...
warning: You appear to have cloned an empty repository.
[admazurskiy@admazurskiy ~]$ ls
Desktop Documents Downloads git-extended gitflow Music Pictures Public Templates Videos work
[admazurskiy@admazurskiy ~]$ cd git-extended/
[admazurskiy@admazurskiy git-extended]$ touch README.md && echo text > README.md
[admazurskiy@admazurskiy git-extended]$ git add .
[admazurskiy@admazurskiy git-extended]$ git commit -m "first commit"
[main (root-commit) d6dd16e] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
[admazurskiy@admazurskiy git-extended]$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 875 bytes | 875.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:nowherewashere/git-extended.git
 * [new branch]      main -> main

```

Рис. 4.3: Создание репозитория

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

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:nowherewashere/git-extended.git",
  "author": "Aleksandr Mazurskiy amd.mazurskiy@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.4: Конфигурация package.json

Делаю снимок изменений, создаю коммит и отправляю на удаленный репозиторий. (рис. 4.5)

```
[admazurskiy@admazurskiy git-extended]$ git add .
[admazurskiy@admazurskiy 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):
(27) improve information content
? 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 125c6ba] docs: improve information content
 1 file changed, 14 insertions(+)
  create mode 100644 package.json
[admazurskiy@admazurskiy git-extended]$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.16 KiB | 1.16 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:nowherewashere/git-extended.git
  d6dd16e..125c6ba  main -> main
```

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)

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

Рис. 4.6: Использование git-flow

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

```
2 files changed, 5 insertions(+)
create mode 100644 CHANGELOG.md
[admazurskiy@admazurskiy git-extended]$ git flow release finish 1.0.0
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 +++
  README.md    | 1 +
2 files changed, 5 insertions(+)
create mode 100644 CHANGELOG.md
Already on 'main'.
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 +++
  README.md    | 1 +
2 files changed, 5 insertions(+)
create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was ae78e65).

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'

[admazurskiy@admazurskiy git-extended]$ git push -all
error: did you mean '--all' (with two dashes)?
[admazurskiy@admazurskiy 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)

```
Machine View
I
foot

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'

[admazurskiy@admazurskiy git-extended]$ nano package.json
[admazurskiy@admazurskiy git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[admazurskiy@admazurskiy git-extended]$ git add .
[admazurskiy@admazurskiy git-extended]$ git status
On branch release/1.2.3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   CHANGELOG.md
    modified:   package.json

[admazurskiy@admazurskiy 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
? 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):
  (16) update 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 476b6a4] chore(site): update changelog
  2 files changed, 5 insertions(+), 1 deletion(-)
[admazurskiy@admazurskiy 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