

Лабораторная работа №4

Продвинутое использование git

Мухин Тимофей Владимирович

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	9

1 Цель работы

Получение навыков правильной работы с репозиториями git

2 Выполнение лабораторной работы

1. Установим gitflow, nodejs и pnpm

```
tvmmukhin@homepc ~$ sudo dnf install gitflow
[sudo] пароль для sey_chik:
Последняя проверка окончания срока действия метаданных: 0:15:29 назад, Сб 09 мар 2024 11:17:12.
Пакет gitflow-1.12.3-1.fc34.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
tvmmukhin@homepc ~$ sudo dnf install nodejs pnpm
Последняя проверка окончания срока действия метаданных: 0:15:51 назад, Сб 09 мар 2024 11:17:12.
Пакет nodejs-1:20.10.0-3.fc39.x86_64 уже установлен.
Пакет pnpm-8.12.0-1.fc39.noarch уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

2. Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.

```
tvmmukhin@homepc ~$ pnpm setup
Appended new lines to /home/sey_chik/.zshrc

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

To start using pnpm, run:
source /home/sey_chik/.zshrc
```

3. Данная программа(commitizen) используется для помощи в форматировании КОММИТОВ

```
tvmmukhin@homepc ~$ pnpm add -g commitizen

Update available! 8.12.0 -> 8.15.4.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v8.15.4
Run "pnpm add -g pnpm" to update.
Follow @pnpmjs for updates: https://twitter.com/pnpmjs

Packages: +152
Progress: resolved 152, reused 0, downloaded 152, added 152, done
Downloading registry.npmjs.org/typescript/5.4.2: 5,82 MB/5,82 MB, done
/home/sey_chik/.local/share/pnpm/global/5:
+ commitizen 4.3.0
Done in 6.3s
```

4. Данная программа (standard-changelog) используется для помощи в создании ЛОГОВ

```

tvmukhin@homepc ~$ pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done

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

Done in 3.4s

```

5. Создание репозитория git-extended

```

tvmukhin@homepc ~/work$ gh repo create git-extended --public
✓ Created repository Seychik23/git-extended on GitHub
https://github.com/Seychik23/git-extended
tvmukhin@homepc ~/work$

```

6. Делаем первый коммит и выкладываем на github

```

tvmukhin@homepc ~/work/git-extended$ git add .
tvmukhin@homepc ~/work/git-extended$ git commit -m "first commit"
[main (корневой коммит) 8e6328d] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

```

7. Конфигурация общепринятых коммитов

```

tvmukhin@homepc ~/work/git-extended$ pnpm init
Wrote to /home/seychik/work/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"
}

```

8. Таким образом, файл package.json приобретает вид

```

GNU nano 7.2 package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:username/git-extended.git",
  "author": "Name Surname <username@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

9. Добавим новые файлы и выполним коммит

```

tvmukhin@homepc:~/work/git-extended$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: (Use arrow keys)
> feat:      A new feature
  fix:       A bug fix
  docs:      Documentation only changes
  style:     Changes that do not affect the meaning of the code (white-space, formatting, missing sem
etc)
  refactor:  A code change that neither fixes a bug nor adds a feature
  perf:     A code change that improves performance
(Move up and down to reveal more choices)

```

10. Конфигурация git flow. Инициализируем

```

tvmukhin@homepc:~/work/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main] main
Branch name for "next release" development: [develop] 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? []
Hooks and filters directory? [/home/sev_chik/work/git-extended/.git/hooks]

```

11. Загрузим весь репозиторий в хранилище и установим внешнюю ветку как вышестоящую для этой ветки

```

[2] + 14469 suspended git branch
tvmukhin@homepc:~/work/git-extended$ git push --all
Everything up-to-date
tvmukhin@homepc:~/work/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
tvmukhin@homepc:~/work/git-extended$

```

12. Создадим релиз с версией 1.0.0

```

branch 'develop' set up to track 'origin/develop'.
tvmukhin@homepc:~/work/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'

```

13. Создадим журнал изменений, добавим журнал изменений в индекс

```

tvmukhin@homepc:~/work/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
tvmukhin@homepc:~/work/git-extended$ git add CHANGELOG.md
tvmukhin@homepc:~/work/git-extended$ git commit -am 'add changelog'
[release/1.0.0 6776950] add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md

```

14. Зальём релизную ветку в основную ветку

```

create mode 100644 CHANGELOG.md
tvmukhin@homepc:~/work/git-extended$ git flow release finish

```

15. Отправим данные на github

```

tvmukhin@homepc:~/work/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 661 байт | 661.00 КиБ/с, готово.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:Seychik23/git-extended.git
   afc18ea..4e72cfe develop -> develop
   afc18ea..4294705 main -> main
tvmukhin@homepc:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байт | 155.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Seychik23/git-extended.git
 * [new tag]         1.0.0 -> 1.0.0

```

16. Создадим релиз на github. Для этого будем использовать утилиты работы с github

```

* [new tag]         1.0.0 -> 1.0.0
tvmukhin@homepc:~/work/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Seychik23/git-extended/releases/tag/v1.0.0
tvmukhin@homepc:~/work/git-extended$

```

17. Создадим ветку для новой функциональности

```

tvmukhin@homepc:~/work/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

```

18. По окончании разработки новой функциональности следующим шагом следует объединить ветку

```

git flow release finish 1.2.3

tvmukhin@homepc:~/work/git-extended$ nano package.json
tvmukhin@homepc:~/work/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
tvmukhin@homepc:~/work/git-extended$ git add CHANGELOG.md
tvmukhin@homepc:~/work/git-extended$ git commit 'update changelog'
error: pathspec 'update changelog' did not match any file(s) known to git
tvmukhin@homepc:~/work/git-extended$ git commit -am 'update changelog'
[release/1.2.3 21217ea] update changelog
2 files changed, 5 insertions(+), 1 deletion(-)

```

19. Обновим номер версии в package.json, создадим журнал изменений, добавим в индекс и отправим данные на сервер

```

tvmukhin@homepc:~/work/git-extended$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 722 байта | 722.00 КиБ/с, готово.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Seychik23/git-extended.git
   4e72cff..8bfd4ee  develop -> develop
   4294705..82dc98d  main -> main
tvmukhin@homepc:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 157 байтов | 157.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Seychik23/git-extended.git
 * [new tag]         1.2.3 -> 1.2.3

```

20. Создадим релиз на github с комментарием из журнала изменений

```

* [new tag]         1.2.3 -> 1.2.3
tvmukhin@homepc:~/work/git-extended$ gh release create v1.2.3

```


3 Выводы

В ходе выполнения работы были изучены новые инструменты git