

Отчёт по лабораторной работе №4

Простейший вариант

Янушкевич Михаил Денисович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

Список иллюстраций

Список таблиц

1 Цель работы

1. Получить навыки правильной работы с репозиториями git.

2 Выполнение лабораторной работы

1. Устанавливаю git-flow.

```
[mdyanushkevich@mdyanushkevich ~]$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:15 назад, Сб 09 мар 2024 19:05:51.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия                Репозиторий  Разме
=====
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 K
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
```

2. Устанавливаю Node.js.

```
[mdyanushkevich@mdyanushkevich ~]$ sudo dnf copr enable elegos/gitflow
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
и 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 репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
[mdyanushkevich@mdyanushkevich ~]$ dnf install gitflow
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на больш
- под именем пользователя root).
[mdyanushkevich@mdyanushkevich ~]$ sudo dnf install gitflow
```

3. Настраиваю Node.js.

```
mdyanushkevich@mdyanushkevich ~]$ sudo apt-get install pnpm
```

4. Устанавливаю программы для форматирования коммитов и создания логов.

```
[mdyanushkevich@mdyanushkevich ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mdyanushkevich/.ssh/id_rsa):
Created directory '/home/mdyanushkevich/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mdyanushkevich/.ssh/id_rsa
Your public key has been saved in /home/mdyanushkevich/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/FAyf974SgqUpGgBfluhhu9LZYLcRLDamCFPYr+VOWU mdyanushkevich@mdyanushkevich
The key's randomart image is:
+---[RSA 4096]-----+
|  o.. .               |
|  . = . .             |
| +, = * .E+ .         |
| o@.B ==+ *           |
| + =.B+= S . .        |
| oo+.. o o o          |
| .o . . . + .         |
| . . . . o .          |
| . . . . .            |
+-----[SHA256]-----+
[mdyanushkevich@mdyanushkevich ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/mdyanushkevich/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
```

5. Создаю репозиторий github.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).


Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * **Repository name ***

 MikePlim /

git-extended is available.

Great repository names are short and memorable. Need inspiration? How about [cuddly-parakeet](#)?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

6. Конфигурирую общепринятые коммиты.

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

7. Редактирую файл package.json.


```

"name": "git-extended",
"version": "1.0.0",
"description": "Git repo for educational pupposes",
"main": "index.js",
"keywords": [],
"repository": "git@github.com:MikePlim/git-extended.git",
"author": "Mikhail Yanushkevich mikhail_2005@mail.ru",
"license": "CC-BY-4.0",
"config": {
  "commitizen": {
    "path": "cz-conventional-changelog"
  }
}

```

8. Добавляю новые файлы, выполняю коммит, отправляю на github.

```

mdyanushkevich@mdyanushkevich git-extended]$ git add .
mdyanushkevich@mdyanushkevich git-extended]$ git cz
z-cli@4.3.0, cz-conventional-changelog@3.3.0

Select the type of change that you're committing: feat:
What is the scope of this change (e.g. component or file)?
Write a short, imperative tense description of the change:
(10) added file
Provide a longer description of the change: (press enter to continue)

Are there any breaking changes? No
Does this change affect any open issues? No
main 8287d11] feat(readme.md): added file
1 file changed, 15 insertions(+)
create mode 100644 package.json
mdyanushkevich@mdyanushkevich git-extended]$ git push

```

9. Инициализирую git-flow, проверяю ветку, загружаю репозиторий в хранилище, устанавливаю внешнюю ветку.

```

[mdyanushkevich@mdyanushkevich 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/mdyanushkevich/git-extended/.git/hooks]
[mdyanushkevich@mdyanushkevich git-extended]$ git branch
* develop
  main
[mdyanushkevich@mdyanushkevich git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/MikePlim/git-extended/pull/new/develop
remote:
To github.com:MikePlim/git-extended.git
 * [new branch]      develop -> develop
[mdyanushkevich@mdyanushkevich git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.

```

10. Создаю релиз с версией 1.0.0, создаю журнал изменений, добавляю журнал в индекс, добавляю релизную ветку в основную, отправляю на github.

```

[mdyanushkevich@mdyanushkevich 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'

[mdyanushkevich@mdyanushkevich git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[mdyanushkevich@mdyanushkevich git-extended]$ git add CHANGELOG.md
[mdyanushkevich@mdyanushkevich git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 cde0fa6] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[mdyanushkevich@mdyanushkevich git-extended]$

```

11. Разрабатываю новую функциональность.

```
[mdyanushkevich@mdyanushkevich git-extended]$ git flow featu
Переключились на новую ветку «feature/feature_branch»

Summary of actions:
- A new branch 'feature/feature_branch' was created, based o
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch

[mdyanushkevich@mdyanushkevich git-extended]$ git flow featu
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была bde2643).
```

12. Создаю релиз git-flow: создаю новый релиз, журнал изменений, добавляю релизную ветку, отправляю данные на github, создаю релиз на github с комментарием.

```
[mdyanushkevich@mdyanushkevich git-extended]$ git
Переключились на новую ветку «release/1.2.3»

Summary of actions:
- A new branch 'release/1.2.3' was created, based
- You are now on branch 'release/1.2.3'

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

    git flow release finish '1.2.3'

[mdyanushkevich@mdyanushkevich git-extended]$ nar
[mdyanushkevich@mdyanushkevich git-extended]$ sta
✓ output changes to CHANGELOG.md
[mdyanushkevich@mdyanushkevich git-extended]$ git
[mdyanushkevich@mdyanushkevich git-extended]$ git
[release/1.2.3 b8cab601 chore(site): update chang
```

3 Выводы

В лабораторной работе №4 я освоил навыки по продвинутому использованию и настройке репозитория git.