

Лабораторная работа №4

Продвинутое использование git

Меркуля Я. А.

2025

Российский университет дружбы народов, Москва, Россия

Вводная часть

- Научиться правильно работать с git flow

Выполнение работы

- Были установлены следующие программы:
 - gitflow
 - nodeJS
 - pnpm

- Через pnpm были установлены:
 - commitizen
 - standard-changelog

```
Done in 1.1s  
[yamerkulov@yamerkulov ~]$ 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/yamerkulov/.local/share/pnpm/global/5:  
+ standard-changelog 6.0.0  
  
Done in 2.7s
```

Рис. 1: Установка через pnpm

Создание нового репозитория

- Репозиторий был создан через сайт GitHub

1 New repository - Mozilla Firefox 40% 100% 10.0.2.15/24 7% 11% us 12:26

yamerkulov - Thunar New repository - Mozilla Firefox [Browser] foot

os-intro__09.03.03: Лабо New repository

← → ↻ https://github.com/new

⚙ Most Visited 🌐 Fedora Docs 📖 Fedora Magazine 📁 Fedora Project 🗑 User Communities 📁 Red Hat 📁 Free Content

No template ▾

Start your repository with a template repository's contents.

Owner * Repository name *

SealDogg / git-extended

git-extended

Great repository names are short and memorable. Need inspiration? How about **cautious-octo-succotash**?

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.

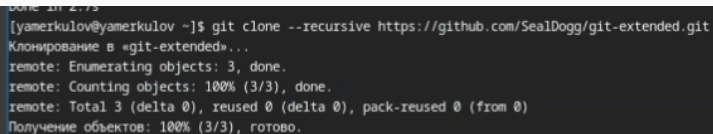
Initialize this repository with:

☒ 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 ▾

- Репозиторий был клонирован с сервера GitHub командой clone



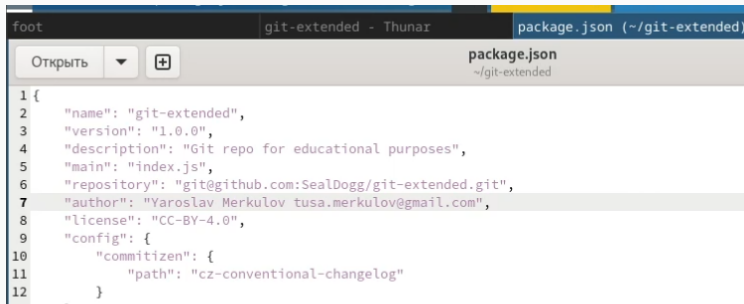
```
done in 2.7s  
[yamerkulov@yamerkulov ~]$ git clone --recursive https://github.com/SealDogg/git-extended.git  
Клонирование в «git-extended»...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Получение объектов: 100% (3/3), готово.
```

Рис. 3: Клонирование репозитория

- После добавления коммита, всё было загружено обратно на сервер

Конфигурация NodeJS

- В файле конфигурации:
 - Вписываем свою фамилию и имя, а также почту
 - Лицензию
 - Формат коммитов
 - Ссылку на репозиторий



The screenshot shows a code editor window with the title bar 'git-extended - Thunar' and a tab 'package.json (~/.git-extended)'. The editor displays the content of the package.json file, which is a JSON object with the following fields: 'name', 'version', 'description', 'main', 'repository', 'author', 'license', and 'config'. The 'author' field is highlighted in the original image. The 'config' field contains a 'commitizen' object with 'path' and 'path' fields.

```
1 {  
2   "name": "git-extended",  
3   "version": "1.0.0",  
4   "description": "Git repo for educational purposes",  
5   "main": "index.js",  
6   "repository": "git@github.com:SealDogg/git-extended.git",  
7   "author": "Yaroslav Merkulov tusa.merkulov@gmail.com",  
8   "license": "CC-BY-4.0",  
9   "config": {  
10    "commitizen": {  
11      "path": "cz-conventional-changelog"  
12    }  
13 }
```

Настройка git flow

- Утилита сама предлагает выбрать настройки, всё выставляем по умолчанию кроме префикса, у нас он будет v

```
[yamerkulov@yamerkulov 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/yamerkulov/git-extended/.git/hooks]
[yamerkulov@yamerkulov git-extended]$
```

Рис. 5: Настройка git flow

- Создаём новый релиз и добавляем туда changelog

```
[yamerkulov@yamerkulov git-extended]$ git branch
* develop
  main
[yamerkulov@yamerkulov 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/SealDogg/git-extended/pull/new/develop
remote:
To https://github.com/SealDogg/git-extended.git
 * [new branch]      develop -> develop
[yamerkulov@yamerkulov git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[yamerkulov@yamerkulov 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:
```

- Аналогично будет создан ещё один релиз 1.2.3
- После второго релиза будет изменена версия в файле конфигурации json
- Все релизы будут загружены на сервер

Выводы

- Были получены и отработаны практические навыки по работа с git flow