

# **Отчёт по лабораторной работе №4**

**Операционные системы**

Пономарева Татьяна Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Установка Node.js . . . . .	8
4.2	Установка gitflow . . . . .	8
4.3	Настройка Node.js . . . . .	8
4.4	Общепринятые коммиты . . . . .	9
4.5	Создание репозитория git . . . . .	9
4.6	Конфигурация общепринятых коммитов . . . . .	10
4.7	Конфигурация git-flow . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1	Установка nodejs и npm . . . . .	8
4.2	Установка gitflow . . . . .	8
4.3	Настройка Node.js . . . . .	9
4.4	Commitizen . . . . .	9
4.5	Standard-changelog . . . . .	9
4.6	Создание git-extended . . . . .	9
4.7	Создание коммита и выкладывание на github . . . . .	10
4.8	Конфигурация для пакетов Node.js . . . . .	10
4.9	package.json . . . . .	11
4.10	Отправление на github . . . . .	11
4.11	Работа с git-flow . . . . .	12
4.12	Загрузка репозитория и установка внешней ветки . . . . .	12
4.13	Создание релиза с версией 1.0.0 . . . . .	12
4.14	Создание журнала изменений и его добавление в индекс . . . . .	13
4.15	Релизная ветка . . . . .	13
4.16	Отправка данных на github . . . . .	14
4.17	Создание релиза на гитхабе . . . . .	14
4.18	Создание новой ветки . . . . .	14
4.19	Файл package.json . . . . .	15
4.20	Создание релиза git-flow . . . . .	15
4.21	Отправка данных на гитхаб . . . . .	16
4.22	Создание релиза на гитхабе с комментарием из журнала изменений . . . . .	16

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git

## 2 Задание

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

## 3 Теоретическое введение

Рабочий процесс Gitflow Workflow используется с пакетом `git-flow`. Gitflow Workflow опубликована и популяризована Винсентом Дриссеном и предполагает выстраивание строгой модели ветвления с учетом выпуска проекта.

## 4 Выполнение лабораторной работы

### 4.1 Установка Node.js

Сначала устанавливаю Node.js и npm (рис. 4.1).

```
[taonomareva@taonomareva ~]$ dnf install nodejs npm
The requested operation requires superuser privileges. Please log in as a user with elevated rights, or use the "--assumeno" or "--downloadonly" options to run the command without modifying the system state.
[taonomareva@taonomareva ~]$ sudo dnf install nodejs npm
[sudo] password for taonomareva:
Updating and loading repositories:
Fedora 41 - x86_64 - Updates 100% | 45.9 KiB/s | 23.1 KiB | 00m01s
Fedora 41 - x86_64 - Updates 100% | 1.4 MiB/s | 2.5 MiB | 00m02s
Repositories loaded
```

Рис. 4.1: Установка nodejs и npm

### 4.2 Установка gitflow

Потом загружаю gitflow (рис. 4.2).

```
[taonomareva@taonomareva ~]$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/develop/contrib/gitflow-installer.sh
er.sh
[taonomareva@taonomareva ~]$ ls
Desktop Documents Downloads gitflow-installer.sh 'Linux version' Music Pictures Public Templates Videos work
[taonomareva@taonomareva ~]$ sudo bash gitflow-installer.sh install stable
[sudo] password for taonomareva:
## git-flow no-make installer ##
Installing git-flow to /usr/local/bin
Cloning repo from Github to gitflow
Cloning into 'gitflow'...
remote: Enumerating objects: 4270, done.
Receiving objects: 0% (1/4270)
```

Рис. 4.2: Установка gitflow

### 4.3 Настройка Node.js

Далее идет настройка Node.js. Запускаю npm, делаю перелогин и выполняю `source ~/.bashrc` (рис. 4.3).



```
[taponomareva@taponomareva ~]$ pnpm setup
No changes to the environment were made. Everything is already up to date.
[taponomareva@taponomareva ~]$ source .bashrc
```

Рис. 4.3: Настройка Node.js

## 4.4 Общепринятые коммиты

Использую commitizen для форматирования коммитов (рис. 4.4).

```
[taponomareva@taponomareva ~]$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 151, reused 151, downloaded 0, added 0, done
Done in 7.9s using pnpm v10.6.1
```

Рис. 4.4: Commitizen

Использую standard-changelog для создания логов (рис. 4.5).

```
[taponomareva@taponomareva ~]$ 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/taponomareva/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0
Done in 6s using pnpm v10.6.1
```

Рис. 4.5: Standard-changelog

## 4.5 Создание репозитория git

Создаю репозиторий git-extended и копирую его в папку home (рис. 4.6).

```
[taponomareva@taponomareva ~]$ gh repo create --public git-extended
✓ Created repository taponomareva/git-extended on GitHub
https://github.com/taponomareva/git-extended
[taponomareva@taponomareva ~]$ git clone --recursive git@github.com:taponomareva/git-extended.git
Cloning into 'git-extended'...
warning: You appear to have cloned an empty repository.
[taponomareva@taponomareva ~]$ ls
Desktop  Documents  Downloads  git-extended  gitflow  'Linux version'  Music  Pictures  Public  Templates  Videos  work
```

Рис. 4.6: Создание git-extended

Делаю первый коммит и выкладываю его на github (рис. 4.7).

```

[taonomareva@taonomareva ~]$ cd git-extended/
[taonomareva@taonomareva git-extended]$ touch README.md && echo text > README.md
[taonomareva@taonomareva git-extended]$ git add .
[taonomareva@taonomareva git-extended]$ git commit -m "first commit"
[main (root-commit) a0eaaae] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
[taonomareva@taonomareva git-extended]$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 874 bytes | 874.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:taonomareva/git-extended.git
 * [new branch]      main -> main

```

Рис. 4.7: Создание коммита и выкладывание на github

## 4.6 Конфигурация общепринятых коммитов

Использую команду `pnpm init`, затем `nano package.json` (рис. 4.8)

```

[taonomareva@taonomareva git-extended]$ pnpm init
Wrote to /home/taonomareva/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",
  "packageManager": "pnpm@10.6.1"
}
[taonomareva@taonomareva git-extended]$ nano package.json

```

Рис. 4.8: Конфигурация для пакетов Node.js

Изменяю файл `package.json` (рис. 4.9).

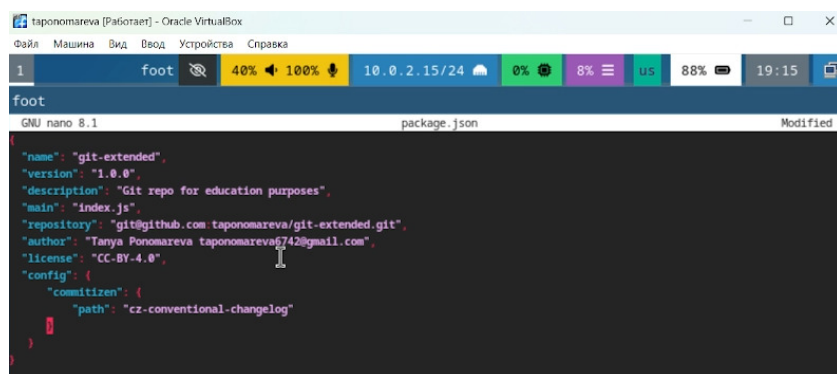


Рис. 4.9: package.json

Добавляю новые файлы, выполняю коммит и отправляю на github (рис. 4.10).

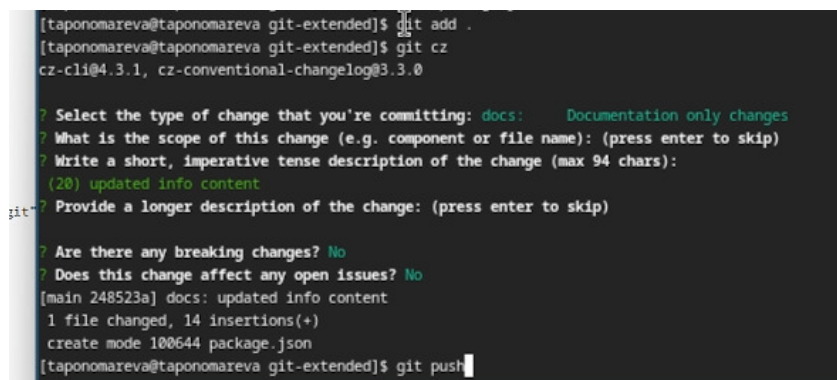


Рис. 4.10: Отправление на github

## 4.7 Конфигурация git-flow

Инициализирую git-flow, устанавливаю префикс для ярлыков как v, проверяю, что нахожусь на ветке develop (рис. 4.11).

```
[taonomareva@taonomareva 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/taonomareva/git-extended/.git/hooks]
[taonomareva@taonomareva git-extended]$ git branch
* develop
main
```

Рис. 4.11: Работа с git-flow

Загружаю весь репозиторий в хранилище и устанавливаю внешнюю ветку как вышестоящую для этой ветки (рис. 4.12).

```
[taonomareva@taonomareva git-extended]$ git push -all
error: did you mean '--all' (with two dashes)?
[taonomareva@taonomareva 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/taonomareva/git-extended/pull/new/develop
remote:
To github.com:taonomareva/git-extended.git
 * [new branch]   develop -> develop
[taonomareva@taonomareva git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 4.12: Загрузка репозитория и установка внешней ветки

Создаю релиз с версией 1.0.0 (рис. 4.13).

```
[taonomareva@taonomareva git-extended]$ git flow release start 1.0.0
Switched to a new branch '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'

[taonomareva@taonomareva git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 4.13: Создание релиза с версией 1.0.0

Создаю журнал изменений и добавляю его в индекс(рис. 4.14).

```

[taonomareva@taonomareva git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[taonomareva@taonomareva git-extended]$ git add .
[taonomareva@taonomareva 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/1.0.0 de840dc] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md

```

Рис. 4.14: Создание журнала изменений и его добавление в индекс

Заливаю релизную ветку в основную (рис. 4.15).

```

[taonomareva@taonomareva git-extended]$ git flow release finish 1.0.0
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was de840dc).

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'

[taonomareva@taonomareva git-extended]$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.69 KiB | 2.69 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:taonomareva/git-extended.git
248523a..33fdd56 develop -> develop

```

Рис. 4.15: Релизная ветка

Отправляю данные на github (рис. 4.16).

```
[taonomareva@taonomareva git-extended]$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.69 KiB | 2.69 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:taonomareva/git-extended.git
 248523a..33fdd56 develop -> develop
 248523a..8b6d958 main -> main
[taonomareva@taonomareva git-extended]$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 166 bytes | 166.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:taonomareva/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
```

Рис. 4.16: Отправка данных на github

Создаю релиз на гитхаб (рис. 4.17).

```
[taonomareva@taonomareva git-extended]$ gh release create v.1.0.0 -F CHANGELOG.md
https://github.com/taonomareva/git-extended/releases/tag/v.1.0.0
```

Рис. 4.17: Создание релиза на гитхабе

Создаю ветку для новой функциональности, затем создаю релиз 1.2.3 (рис. 4.18).

```
[taonomareva@taonomareva git-extended]$ git flow feature start feature_branch
Switched to a new branch '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

[taonomareva@taonomareva git-extended]$ git branch
  develop
 * feature/feature_branch
    main
[taonomareva@taonomareva git-extended]$ git flow release start 1.2.3
```

Рис. 4.18: Создание новой ветки

Обновляю номер версии в файле package.json (рис. 4.19).

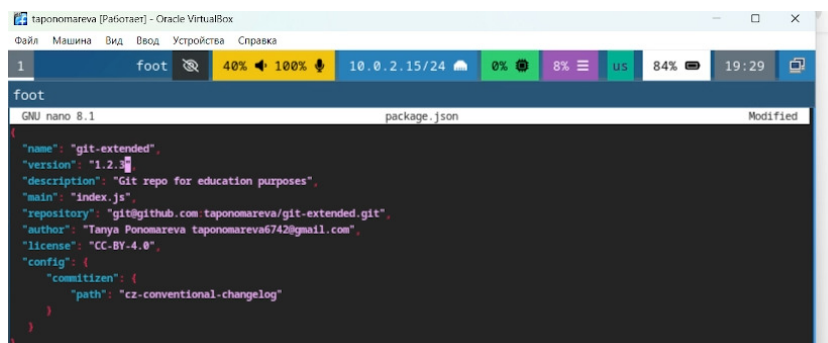


Рис. 4.19: Файл package.json

Создаю журнал изменений, добавляю его в индекс и заливаю релизную ветку в основную (рис. 4.20).

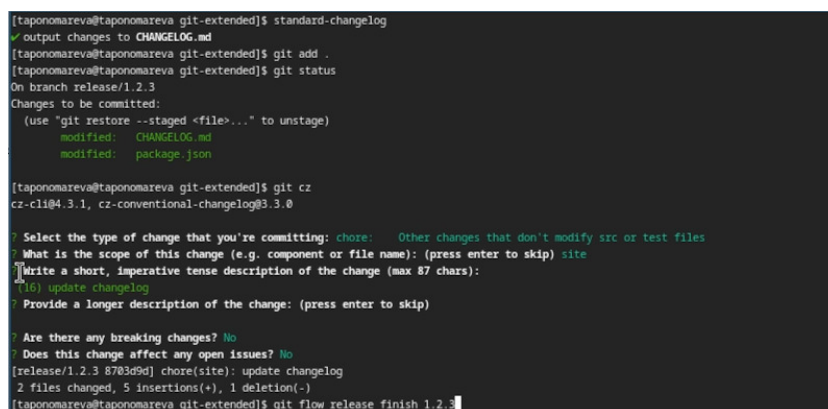


Рис. 4.20: Создание релиза git-flow

Отправляю данные на гитхаб (рис. 4.21).

```

[taponomareva@taponomareva git-extended]$ git push --all
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 6 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.80 KiB | 1.40 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:taponomareva/git-extended.git
   33fdd56..b959b42  develop -> develop
   8b6d958..60119bd  main -> main
   [new branch]      feature/feature_branch -> feature/feature_branch
[taponomareva@taponomareva git-extended]$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 165 bytes | 165.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:taponomareva/git-extended.git
   * [new tag]         v1.2.3 -> v1.2.3

```

Рис. 4.21: Отправка данных на гитхаб

Создаю релиз на гитхаб с комментарием из журнала изменений (рис. 4.22).

```

[taponomareva@taponomareva git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/taponomareva/git-extended/releases/tag/v1.2.3

```

Рис. 4.22: Создание релиза на гитхабе с комментарием из журнала изменений



## **5 Выводы**

Были получены навыков правильной работы с репозиториями git

# **Список литературы**

1. Курс на ТУИС