

# **Отчёт по лабораторной работе №4**

**Продвинутое использование git**

Диденко Дмитрий Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Работа с тестовым репозиторием . . . . .	6
3.2	Подготовка рабочего репозитория . . . . .	12
<b>4</b>	<b>Вывод</b>	<b>14</b>

# Список иллюстраций

3.1	Node.js . . . . .	6
3.2	установка commitizen . . . . .	7
3.3	установка standard-changelog . . . . .	7
3.4	package.json . . . . .	8
3.5	Отправка . . . . .	8
3.6	Инициализация и загрузка изменений . . . . .	9
3.7	Завершение релиза . . . . .	10
3.8	Отправка . . . . .	10
3.9	Объединение веток . . . . .	11
3.10	Завершение релиза . . . . .	12
3.11	package.json и коммит . . . . .	12
3.12	Завершение релиза . . . . .	13

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

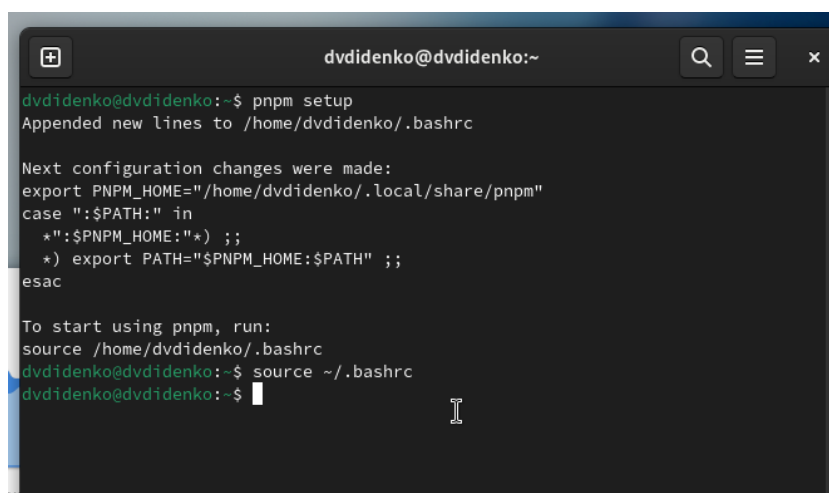
## 2 Теоретические сведения

- 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.
- Последовательность действий при работе по модели Gitflow:
- Из ветки master создаётся ветка develop.
- Из ветки develop создаётся ветка release.
- Из ветки develop создаются ветки feature.
- Когда работа над веткой feature завершена, она сливается с веткой develop.
- Когда работа над веткой релиза release завершена, она сливается в ветки develop и master.
- Если в master обнаружена проблема, из master создаётся ветка hotfix.
- Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

## 3 Выполнение лабораторной работы

### 3.1 Работа с тестовым репозиторием

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

A terminal window with a dark background and light-colored text. The window title is 'dvdidenko@dvdidenko:~'. The terminal shows the output of 'pnpm setup', which includes instructions on how to configure the environment by adding lines to the .bashrc file. The user then runs 'source ~/.bashrc' to apply the changes. The terminal text is as follows:

```
dvdidenko@dvdidenko:~$ pnpm setup
Appended new lines to /home/dvdidenko/.bashrc

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

To start using pnpm, run:
source /home/dvdidenko/.bashrc
dvdidenko@dvdidenko:~$ source ~/.bashrc
dvdidenko@dvdidenko:~$
```

Рис. 3.1: Node.js

Программа commitizen используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов.

```
dvdidenko@dvdidenko:~$  
dvdidenko@dvdidenko:~$ pnpm add -g commitizen  
  
Update available! 8.12.0 → 9.1.4.  
Changelog: https://github.com/pnpm/pnpm/releases/tag/v9.1.4  
Run "pnpm add -g pnpm" to update.  
  
Follow @pnpmjs for updates: https://twitter.com/pnpmjs  
  
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6  
Packages: +153  
+-----+  
Downloading registry.npmjs.org/typescript/5.4.5: 5,83 MB/5,83 MB, done  
Progress: resolved 153, reused 0, downloaded 153, added 153, done  
  
/home/dvdidenko/.local/share/pnpm/global/5:  
+ commitizen 4.3.0  
  
Done in 7.6s  
dvdidenko@dvdidenko:~$
```

Рис. 3.2: установка commitizen

Программа standard-changelog используется для помощи в создании логов.

```
dvdidenko@dvdidenko:~$  
dvdidenko@dvdidenko:~$ pnpm add -g standard-changelog  
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6  
Packages: +42  
+-----+  
Progress: resolved 195, reused 153, downloaded 42, added 42, done  
  
/home/dvdidenko/.local/share/pnpm/global/5:  
+ standard-changelog 6.0.0  
  
Done in 5.5s  
dvdidenko@dvdidenko:~$
```

Рис. 3.3: установка standard-changelog

Делаем первый коммит и выкладываем на github.

Необходимо заполнить несколько параметров пакета.

Таким образом, файл package.json приобретает вид:

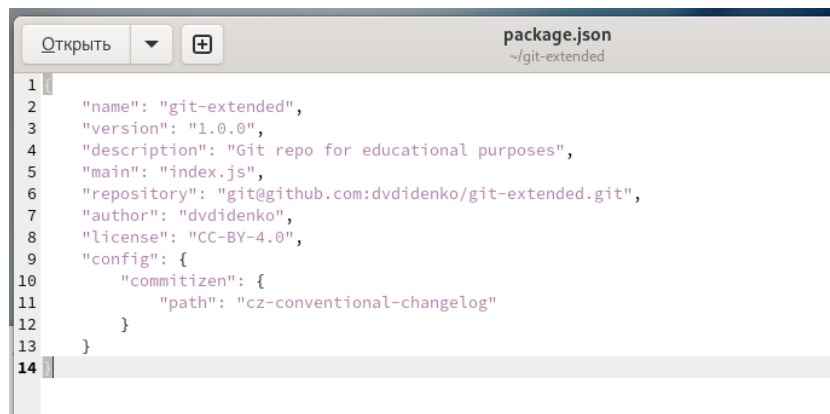


Рис. 3.4: package.json

Добавим новые файлы.

Выполним коммит.

Отправим на github.

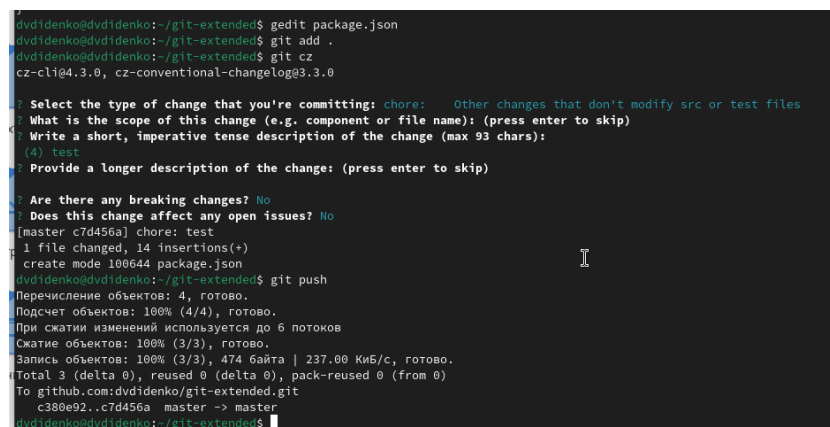


Рис. 3.5: Отправка

Инициализируем git-flow

Проверьте, что Вы на ветке develop

Загрузите весь репозиторий в хранилище



```

dvdidenko@dvdidenko:~/git-extended$
dvdidenko@dvdidenko:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
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/dvdidenko/git-extended/.git/hooks]
dvdidenko@dvdidenko:~/git-extended$ git branch
* develop
  master
dvdidenko@dvdidenko:~/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/dvdidenko/git-extended/pull/new/develop
remote:
To github.com:dvdidenko/git-extended.git
* [new branch]      develop -> develop
dvdidenko@dvdidenko:~/git-extended$

```

Рис. 3.6: Инициализация и загрузка изменений

Установите внешнюю ветку как вышестоящую для этой ветки

Создадим релиз с версией 1.0.0

Создадим журнал изменений

Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

```
dvddenko@dvddenko:~/git-extended
[release/1.0.0 8fd667d] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
dvddenko@dvddenko:~/git-extended$ git flow release finish 1.0.0
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
dvddenko@dvddenko:~/git-extended$ git flow release finish 1.0.0
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 8fd667d).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- 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'

dvddenko@dvddenko:~/git-extended$
```

Рис. 3.7: Завершение релиза

## Отправим данные на github

```
dvddenko@dvddenko:~/git-extended$
dvddenko@dvddenko:~/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 550 байтов | 183.00 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:dvddenko/git-extended.git
 c7d456a..dc10543 develop -> develop
 c7d456a..d18edae master -> master
dvddenko@dvddenko:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байтов | 155.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dvddenko/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
dvddenko@dvddenko:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/dvddenko/git-extended/releases/tag/v1.0.0
dvddenko@dvddenko:~/git-extended$
```

Рис. 3.8: Отправка

Создадим ветку для новой функциональности По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch с develop:

```
dvdidenko@dvdidenko: ~/git-extended$
dvdidenko@dvdidenko:~/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

dvdidenko@dvdidenko:~/git-extended$ touch test2
dvdidenko@dvdidenko:~/git-extended$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была dc10543).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

dvdidenko@dvdidenko:~/git-extended$
```

Рис. 3.9: Объединение веток

Создадим релиз с версией 1.2.3

Обновите номер версии в файле package.json. Установите её в 1.2.3

Создадим журнал изменений

Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

```
dvddenko@dvddenko:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 3e2b27b] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
dvddenko@dvddenko:~/git-extended$ git flow release finish 1.2.3
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «master»
Ваша ветка опережает «origin/master» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
dvddenko@dvddenko:~/git-extended$ git flow release finish 1.2.3
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Уже на «master»
Ваша ветка опережает «origin/master» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 3e2b27b).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'master'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'
dvddenko@dvddenko:~/git-extended$
```

Рис. 3.10: Завершение релиза

## 3.2 Подготовка рабочего репозитория

```
./
"keywords": [],
"author": "",
"license": "ISC"
}
dvddenko@dvddenko:~/work/study/2023-2024/Операционные системы/os-intro$ gedit package.json
dvddenko@dvddenko:~/work/study/2023-2024/Операционные системы/os-intro$ git add .
dvddenko@dvddenko:~/work/study/2023-2024/Операционные системы/os-intro$ git cz
cz-cli@4.3.0, 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)
? Write a short, imperative tense description of the change (max 93 chars):
(4) test
? 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
[master 63b40c2] chore: test
71 files changed, 2374 insertions(+), 730 deletions(-)
create mode 160544 labs/Lab01/01_25n
```

Рис. 3.11: package.json и коммит

```
dvdidenko@dvdidenko:~/work/study/2023-2024/Операционные системы/os-intro$
dvdidenko@dvdidenko:~/work/study/2023-2024/Операционные системы/os-intro$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
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/dvdidenko/work/study/2023-2024/Операционные системы/os-intro/.git/hooks]
dvdidenko@dvdidenko:~/work/study/2023-2024/Операционные системы/os-intro$ git branch
* develop
  master
dvdidenko@dvdidenko:~/work/study/2023-2024/Операционные системы/os-intro$
```

Рис. 3.12: Завершение релиза

## 4 Вывод

Мы приобрели практические навыки взаимодействия с дополнительными функциями гитхаб.