

Отчёт по лабораторной работе №4

Продвинутое использование git

Матиев Даниэль Саматович

Содержание

1 Цель работы	5
2 Теоретические сведения	6
3 Выполнение лабораторной работы	8
3.1 Работа с тестовым репозиторием	8
3.2 Подготовка рабочего репозитория	19
4 Вывод	21

Список иллюстраций

3.1 Node.js	9
3.2 установка commitizen	10
3.3 установка standard-changelog	11
3.4 package.json	12
3.5 Отправка	13
3.6 Инициализация и загрузка изменений	14
3.7 Завершение релиза	15
3.8 Отправка	16
3.9 Объединение веток	17
3.10 Завершение релиза	18
3.11 package.json и коммит	19
3.12 Завершение релиза	20

Список таблиц

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 слиивается с веткой develop.
 - Когда работа над веткой релиза release завершена, она слиивается в ветки develop и master.
 - Если в master обнаружена проблема, из master создаётся ветка hotfix.

- Когда работа над веткой исправления hotfix завершена, она сливаются 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.

```
dsmatiev@dsmatiev:~$  
dsmatiev@dsmatiev:~$ pnpm setup  
Appended new lines to /home/dsmatiev/.bashrc [ ]  
  
Next configuration changes were made:  
export PNPM_HOME="/home/dsmatiev/.local/share/pnpm"  
case ":$PATH:" in  
  *":$PNPM_HOME:") ;;  
  *) export PATH="$PNPM_HOME:$PATH" ;;  
esac  
  
To start using pnpm, run:  
source /home/dsmatiev/.bashrc  
dsmatiev@dsmatiev:~$ source ~/.bashrc  
dsmatiev@dsmatiev:~$ [ ]
```

Рисунок 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.

```
dsmatiev@dsmatiev:~$ pnpm add -g commitizen

Update available! 10.27.0 → 10.30.2.
Changelog: https://pnpm.io/v/10.30.2
To update, run: pnpm self-update

[

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +149
+++++
Progress: resolved 149, reused 0, downloaded 149, added 149, done

/home/dsmatiev/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 2.6s using pnpm v10.27.0
dsmatiev@dsmatiev:~$ ]
```

Рисунок 3.2: установка commitizen

Программа standard-changelog используется для помощи в создании логов.

```
dsmatiev@dsmatiev:~$ pnpm add -g standard-changelog
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +32
+++++
Progress: resolved 181, reused 149, downloaded 32, added 32, done

/home/dsmatiev/.local/share/pnpm/global/5:
+ standard-changelog 7.0.1

Done in 3.1s using pnpm v10.27.0
dsmatiev@dsmatiev:~$
```

Рисунок 3.3: установка standard-changelog

Делаем первый коммит и выкладываем на github.

Необходимо заполнить несколько параметров пакета.

Таким образом, файл package.json приобретает вид:



The screenshot shows a code editor window with the title bar "package.json" and the path "~git-extended". The editor displays the following JSON code:

```
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:dsmatiev/git-extended.git",
7   "author": "dsmatiev",
8   "license": "CC-BY-4.0",
9   "config": {
10     "commitizen": {
11       "path": "cz-conventional-changelog"
12     }
13   }
14 }
```

Рисунок 3.4: package.json

Добавим новые файлы.

Выполним коммит.

Отправим на github.

```
dsmatiev@dsmatiev:~/git-extended$ git add .
dsmatiev@dsmatiev:~/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)
? 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 8e4fe20] chore: test
 1 file changed, 14 insertions(+)
 create mode 100644 package.json
dsmatiev@dsmatiev:~/git-extended$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.11 KiB | 1.11 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dsmatiev/git-extended.git
 43f6358..8e4fe20  master -> master
dsmatiev@dsmatiev:~/git-extended$
```

Рисунок 3.5: Отправка

Инициализируем git-flow

Проверьте, что Вы на ветке develop

Загрузите весь репозиторий в хранилище

```
dsmatiev@dsmatiev:~/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/dsmatiev/git-extended/.git/hooks]
dsmatiev@dsmatiev:~/git-extended$ git branch
* develop
  master

dsmatiev@dsmatiev:~/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/dsmatiev/git-extended/pull/new/develop
remote:
To github.com:dsmatiev/git-extended.git
 * [new branch]    develop -> develop
dsmatiev@dsmatiev:~/git-extended$
```

Рисунок 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

```
dsmatiev@dsmatiev:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
dsmatiev@dsmatiev:~/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'

dsmatiev@dsmatiev:~/git-extended$ standard-changelog --first-release
dsmatiev@dsmatiev:~/git-extended$ git add CHANGELOG.md
dsmatiev@dsmatiev:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 d3a1b37] chore(site): add changelog
  1 file changed, 4 insertions(+)
  create mode 100644 CHANGELOG.md
dsmatiev@dsmatiev:~/git-extended$ git flow release finish 1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 +++
  1 file changed, 4 insertions(+)
  create mode 100644 CHANGELOG.md
Already on 'master'
Your branch is ahead of 'origin/master' by 2 commits
```

Рисунок 3.7: Завершение релиза

Отправим данные на github

```
dsmatiev@dsmatiev:~/git-extended$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.66 KiB | 2.66 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dsmatiev/git-extended.git
  8e4fe20..1c6104f develop -> develop
  8e4fe20..4f5a822 master -> master
dsmatiev@dsmatiev:~/git-extended$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 153 bytes | 153.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dsmatiev/git-extended.git
 * [new tag]      v1.0.0 -> v1.0.0
dsmatiev@dsmatiev:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/dsmatiev/git-extended/releases/tag/v1.0.0
dsmatiev@dsmatiev:~/git-extended$
```

Рисунок 3.8: Отправка

Создадим ветку для новой функциональности По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop:

```
dsmatiev@dsmatiev:~/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
```

```
dsmatiev@dsmatiev:~/git-extended$ touch test2  
dsmatiev@dsmatiev:~/git-extended$ git flow feature finish feature_branch  
Switched to branch 'develop'  
Your branch is up to date with 'origin/develop'.  
Already up to date.  
Deleted branch feature/feature_branch (was 1c6104f).
```

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'

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

Рисунок 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

```
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'

dsmatiev@dsmatiev:~/git-extended$ git push --all
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.77 KiB | 2.77 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:dsmatiev/git-extended.git
  1c6104f..22e300c  develop -> develop
  4f5a822..df8e16a  master -> master
dsmatiev@dsmatiev:~/git-extended$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 154 bytes | 154.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dsmatiev/git-extended.git
 * [new tag]      v1.2.3 -> v1.2.3
dsmatiev@dsmatiev:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/dsmatiev/git-extended/releases/tag/v1.2.3
dsmatiev@dsmatiev:~/git-extended$
```

Рисунок 3.10: Завершение релиза

3.2 Подготовка рабочего репозитория

```
dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ 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)  
? 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 d25b7d6] chore: test  
 1 file changed, 11 insertions(+), 11 deletions(-)  
dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ git pu  
sh  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 1.10 KiB | 1.10 MiB/s, done.  
Total 3 (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:dsmatiev/2026-1--study--os-intro.git  
 6931913..d25b7d6 master -> master  
dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$
```

Рисунок 3.11: package.json и коммит

```
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'

dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ git pu
sh --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.74 KiB | 2.74 MiB/s, done.
Total 5 (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:dsmatiev/2026-1--study--os-intro.git
  d25b7d6..7e80e97  develop -> develop
  d25b7d6..5c6289b  master -> master
dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ git pu
sh --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 154 bytes | 154.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dsmatiev/2026-1--study--os-intro.git
 * [new tag]      v1.0.0 -> v1.0.0
dsmatiev@dsmatiev:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$
```

Рисунок 3.12: Завершение релиза

4 Вывод

Мы приобрели практические навыки взаимодействия с дополнительными функциями гитхаб.