

# **Отчёт по лабораторной работе №4**

**Операционные системы**

Козина Дарья 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>Выполнение лабораторной работы</b>             | <b>7</b>  |
| 3.1      | Установка программного обеспечения . . . . .      | 7         |
| 3.1.1    | Общепринятые коммиты . . . . .                    | 8         |
| 3.2      | Практический сценарий использования git . . . . . | 8         |
| <b>4</b> | <b>Выводы</b>                                     | <b>15</b> |

# Список иллюстраций

|      |                                    |    |
|------|------------------------------------|----|
| 3.1  | Режим супер-пользователя . . . . . | 7  |
| 3.2  | Установка git-flow . . . . .       | 7  |
| 3.3  | Установка Node.js . . . . .        | 7  |
| 3.4  | Установка npm . . . . .            | 7  |
| 3.5  | Запуск . . . . .                   | 8  |
| 3.6  | Команда source . . . . .           | 8  |
| 3.7  | Comitizen и git-cz . . . . .       | 8  |
| 3.8  | Standard-changelog . . . . .       | 8  |
| 3.9  | Создание репозитория . . . . .     | 9  |
| 3.10 | Первый коммит . . . . .            | 9  |
| 3.11 | Конфигурация . . . . .             | 9  |
| 3.12 | package.json . . . . .             | 10 |
| 3.13 | Инициализация . . . . .            | 10 |
| 3.14 | Проверка . . . . .                 | 11 |
| 3.15 | Загрузка репозитория . . . . .     | 11 |
| 3.16 | Установка внешней ветки . . . . .  | 11 |
| 3.17 | Создание релиза . . . . .          | 11 |
| 3.18 | Создание журнала . . . . .         | 12 |
| 3.19 | Создание журнала . . . . .         | 12 |
| 3.20 | Отправка данных . . . . .          | 12 |
| 3.21 | Создание релиза . . . . .          | 12 |
| 3.22 | Создание ветки . . . . .           | 13 |

## **Список таблиц**

# 1 Цель работы

Получить навыки правильной работы с репозиторием git.

## 2 Задание

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

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

Перейдем в режим супер-пользователя (рис. 3.1).

```
dakozina@vbox:~$ sudo -i
[sudo] пароль для dakozina:
root@vbox:~# dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org 100% | 206.0 B/s | 652.0 B | 00m03s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не явл
```

Рис. 3.1: Режим супер-пользователя

Установим git-flow (рис. 3.2).

```
root@vbox:~# dnf install gitflow
Обновление и загрузка репозитория:
Fedora 41 - x86_64 ???% | 0.0 B/s | 0.0 B | 00m00s
```

Рис. 3.2: Установка git-flow

Установим Node.js (рис. 3.3, рис. 3.4).

```
root@vbox:~# dnf install nodejs
Обновление и загрузка репозитория:
```

Рис. 3.3: Установка Node.js

```
root@vbox:~# dnf install npm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
npm noarch 9.13.0-1.fc41 updates 15.5 MiB
```

Рис. 3.4: Установка npm

Настроим Node.js. Запустим (рис. 3.5).

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

Next configuration changes were made:
export Pnpm_HOME="/home/dakozina/.local/share/pnpm"
case "$PATH:" in
  *"$Pnpm_HOME:") ;;
```

Рис. 3.5: Запуск

Выполним команду source (рис. 3.6).

```
dakozina@vbox:~$ source ~/.bashrc
```

Рис. 3.6: Команда source

### 3.1.1 Общепринятые коммиты

Добавим commitizen и git-cz (рис. 3.7).

```
dakozina@vbox:~$ pnpm add -g commitizen

Update available! 9.13.0 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm self-update" to update.

Follow pnpm for updates: https://x.com/pnpmjs
```

Рис. 3.7: Comitizen и git-cz

Добавим standard-changelog (рис. 3.8).

```
dakozina@vbox:~$ pnpm add -g standart-changelog
Progress: resolved 1, reused 1, downloaded 0, added 0
```

Рис. 3.8: Standard-changelog

## 3.2 Практический сценарий использования git

Создадим репозиторий git-extended (рис. 3.9).



```
dakozina@vbox:~$ gh repo create git-extended --public
✓ Created repository dakozina/git-extended on GitHub
https://github.com/dakozina/git-extended
dakozina@vbox:~$ git clone --recursive git@github.com:dakozina/git-extended
Клонирование в «git-extended»...
```

Рис. 3.9: Создание репозитория

Делаем первый коммит и выложим на GitHub (рис. 3.10).

```
dakozina@vbox:~/git-extended$ git commit -m "first commit"
Текущая ветка: master

Начальный коммит

ничего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
dakozina@vbox:~/git-extended$ git remote add origin git@github.com:dakozina/git-extended
error: внешний репозиторий origin уже существует
dakozina@vbox:~/git-extended$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:dakozina/git-extended»
dakozina@vbox:~/git-extended$ git branch -M master
dakozina@vbox:~/git-extended$ git remote add origin git@github.com:dakozina/git-extended
error: внешний репозиторий origin уже существует
dakozina@vbox:~/git-extended$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:dakozina/git-extended»
```

Рис. 3.10: Первый коммит

Конфигурация для пакетов Node.js (рис. 3.11).

```
dakozina@vbox:~/git-extended$ pnpm init
Wrote to /home/dakozina/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"
}
```

Рис. 3.11: Конфигурация

Отредактируем файл package.json (рис. 3.12).

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:username/git-extended.git",
  "author": "Name Surname <username@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.12: package.json

Добавим новые файлы, выполним коммит и отправим на github (рис. ??).

```
dakozina@vbox:~/git-extended$ git add .
dakozina@vbox:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: feat:      A new feature
? 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 94 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 (корневой коммит) b07c556] feat: test
 1 file changed, 14 insertions(+)
 create mode 100644 package.json
dakozina@vbox:~/git-extended$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1.06 КиБ | 1.06 МиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dakozina/git-extended
 * [new branch]      master -> master
dakozina@vbox:~/git-extended$
```

Инициализируем git-flow (рис. 3.13).

```
dakozina@vbox:~/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? []
Hooks and filters directory? [/home/dakozina/git-extended/.git/hooks]
```

Рис. 3.13: Инициализация

Проверим, что мы на ветке develop (рис. 3.14).

```
dakozina@vbox:~/git-extended$ git branch
* develop
  master
```

Рис. 3.14: Проверка

Загрузим весь репозиторий в хранилище (рис. 3.15).

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

Рис. 3.15: Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 3.16).

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

Рис. 3.16: Установка внешней ветки

Создадим релиз с версией 1.0.0 (рис. 3.17).

```
dakozina@vbox:~/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'
```

Рис. 3.17: Создание релиза

Создадим журнал изменений (рис. 3.18).

```
dakozina@vbox:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 3.18: Создание журнала

Создадим журнал изменений в индексе (рис. 3.19).

```
dakozina@vbox:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 8e45576] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.19: Создание журнала

Зальем релизную ветку в основную ветку (рис. ??).

```
dakozina@vbox:~/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 | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
```

Отправим данные на github (рис. 3.20).

```
dakozina@vbox:~/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.90 КиБ | 1.90 МиБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:dakozina/git-extended
 b07c556..08e3634 master -> master
 * [new branch]      release/1.0.0 -> release/1.0.0
dakozina@vbox:~/git-extended$ git push --tags
Everything up-to-date
```

Рис. 3.20: Отправка данных

Создадим релиз на github. Для этошо будем использовать утилиты работы с github (рис. 3.21).

```
dakozina@vbox:~/git-extended$ gh release create x1.0.0 -F CHANGELOG.md
https://github.com/dakozina/git-extended/releases/tag/x1.0.0
```

Рис. 3.21: Создание релиза

Создадим ветку для новой функциональности (рис. 3.22).

```
dakozina@vbox:~/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
```

Рис. 3.22: Создание ветки

Продолжим работу с git как обычно. В конце разработки новой функциональности следующим шагом объединим ветку feature\_branch с develop (рис. ??).

```
dakozina@vbox:~/git-extended$ touch test2
dakozina@vbox:~/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 b07c556).

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'
```

Проделаем все то же самое для нашего основного репозитория (рис. ??, рис. ??, рис. ??, рис. ??, рис. ??).

```
dakozina@vbox:~$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro
dakozina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ npm init
Wrote to /home/dakozina/work/study/2024-2025/Операционные системы/os-intro/package.json

{
  "name": "os-intro",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

{
  "name": "os-intro",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "repository": "git@github.com:dakozina/study_2024-2025_os-intro.git",
  "author": "Daria Kozina <1132246831@pfur.ru>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

```

dakoizina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
dakoizina@vbox:~/work/study/2024-2025/Операционные системы/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 f81ccc2] chore: test
436 files changed, 63355 insertions(+)
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/1.png
create mode 100644 labs/lab01/presentation/image/10.png
create mode 100644 labs/lab01/presentation/image/11.png
create mode 100644 labs/lab01/presentation/image/12.png
create mode 100644 labs/lab01/presentation/image/13.png

dakoizina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 156, готово.
Подсчет объектов: 100% (156/156), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (152/152), готово.
Запись объектов: 43% (67/155), 3.09 МиБ | 930.00 КиБ/с

120e6d4..f81ccc2 master -> master
dakoizina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Everything up-to-date
dakoizina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Everything up-to-date
dakoizina@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ █

```

## 4 Выводы

В ходе лабораторной работы мы получили навыки правильной работы с репозиторием git.