

Лабораторная работа №4

Операционные системы

Краснова Камилла Геннадьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Практический сценарий использования git	9
4	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Установка репозитория сорг	7
3.2	Установка git-flow	7
3.3	Установка node.js	8
3.4	Установка npnm	8
3.5	Запуск npnm	8
3.6	Программа commitizen	9
3.7	Программа standard-changelog	9
3.8	Создание первого коммита	9
3.9	Конфигурация для пакетов Node.js	10
3.10	Заполнение параметров пакета	10
3.11	Добавление новых файлов	11
3.12	Инициализация git-flow	11
3.13	Проверка ветки	11
3.14	Загрузка репозитория в хранилище	11
3.15	Создание релиза	12
3.16	Создание журнала изменений	12
3.17	Добавление журнала изменений в индекс	12
3.18	Релизная ветка	12
3.19	Отправка данных	13
3.20	Отправка данных	13
3.21	Создание релиза	13
3.22	Создание ветки	13
3.23	Объединение ветки	14
3.24	Создание релиза	14
3.25	Изменение файла	14
3.26	Создание журнала изменений	15
3.27	Релизная ветка	15
3.28	Отправка данных	16
3.29	Отправка данных	16
3.30	Создание релиза	16
3.31	Репозиторий	17

Список таблиц

1 Цель работы

Целью данной лабораторной работы является получение навыков правильной работы с репозиториями git.

2 Задание

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

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Для начала устанавливаю репозиторий copr (рис. 3.1).

```
kamilla@fedora:~$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для kamilla:
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build>
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого ре
```

Рис. 3.1: Установка репозитория copr

И устанавливаю сам git-flow (рис. 3.2).

```
kamilla@fedora:~$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos
Copr repo for PyCharm owned by phracek
Fedora 40 - x86_64 - Updates
Fedora 40 - x86_64 - Updates
```

Рис. 3.2: Установка git-flow

Устанавливаю node.js (рис. 3.3).

```

kamilla@fedora:~$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:00:51 назад, С6 0
Зависимости разрешены.
=====
Пакет                Архитектура          Версия
=====
Установка:
nodejs                x86_64                1:20.18.2-2.fc40
Установка зависимостей:
nodejs-libs           x86_64                1:20.18.2-2.fc40
Установка слабых зависимостей:
nodejs-docs           noarch                1:20.18.2-2.fc40
nodejs-full-i18n      x86_64                1:20.18.2-2.fc40
nodejs-npm            x86_64                1:10.8.2-1.20.18.2.2.fc40

```

Рис. 3.3: Установка node.js

И затем устанавливаю pnpm (рис. 3.4).

```

Выполнено:
kamilla@fedora:~$ sudo dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:01:59
Зависимости разрешены.
=====
Пакет                Архитектура          Версия
=====
Установка:
pnpm                 noarch                9.0.6-1.fc40
Результат транзакции
=====
Установка 1 Пакет
Объем загрузки: 2.8 М

```

Рис. 3.4: Установка pnpm

Запускаю pnpm и выполняю 'source ~/.bashrc' (рис. 3.5).

```

Выполнено:
kamilla@fedora:~$ pnpm setup
Appended new lines to /home/kamilla/.bashrc

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

```

Рис. 3.5: Запуск pnpm

Использую программу commitizen для помощи в форматировании коммитов (рис. 3.6).


```
kamilla@fedora:~$ pnpm add -g commitizen

Update available! 9.0.6 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
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: +151
Progress: resolved 151, reused 0, downloaded 151, added 151, done
```

Рис. 3.6: Программа commitizen

Далее использую программу standard-changelog, которая помогает в создании логов (рис. 3.7).

```
kamilla@fedora:~$ 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/kamilla/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0
```

Рис. 3.7: Программа standard-changelog

3.2 Практический сценарий использования git

Создаю репозиторий на GitHub с названием git-extended. Делаю первый коммит и выкладываю на github (рис. 3.8).

```
kamilla@fedora:~/git-extended$ git commit -m "first commit"
Текущая ветка: main

Начальный коммит

нечего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
kamilla@fedora:~/git-extended$ git remote add origin git@github.com:kakras/git-extended.git
error: внешний репозиторий origin уже существует
kamilla@fedora:~/git-extended$ git push -u origin master
error: src refspec master не соответствует никакому объекту
```

Рис. 3.8: Создание первого коммита

Прописываю команду pnpm init, затем перехожу в директорию, в которой находится файл package.json (рис. 3.9).

```

kamilla@fedora:~/git-extended$ pnpm init
Wrote to /home/kamilla/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. 3.9: Конфигурация для пакетов Node.js

С помощью текстового редактора я заполняю несколько параметров пакета. После изменения он выглядит следующим образом: (рис. 3.10).

```

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:kakras/git-extended.git",
7   "author": "Kamilla Krasnova lzip184@gmail.com",
8   "license": "CC-BY-4.0",
9   "config": {
10     "commitizen": {
11       "path": "cz-conventional-changelog"
12     }
13   }
14 }

```

Рис. 3.10: Заполнение параметров пакета

Дальше я добавляю новые файлы и выполняю коммит, после чего отвечаю на пару наводящих вопросов и отправляю все на github (рис. 3.11).

```

kamilla@fedora:~/git-extended$ git add .
kamilla@fedora:~/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
? 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):
(56) this change initializes a new Node.js project using npm

```

Рис. 3.11: Добавление новых файлов

Далее инициализирую git-flow. Префикс для ярлыков устанавливаю в v (рис. 3.12).

```

kamilla@fedora:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] master
Branch name for "next release" development: [develop] 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/kamilla/git-extended/.git/hooks]
kamilla@fedora:~/git-extended$

```

Рис. 3.12: Инициализация git-flow

Проверяю что я на ветке develop (рис. 3.13).

```

kamilla@fedora:~/git-extended$ git branch
* develop
  master

```

Рис. 3.13: Проверка ветки

Загружаю весь репозиторий в хранилище (рис. 3.14).

```

kamilla@fedora:~/git-extended$ git push --all
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 481 байт | 481.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:kakras/git-extended.git
 * [new branch]      develop -> develop
 * [new branch]      master -> master

```

Рис. 3.14: Загрузка репозитория в хранилище

Устанавливаю внешнюю ветку как вышестоящую для этой ветки. Далее создаю релиз с версией 1.0.0 (рис. 3.15).

```
kamilla@fedora:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
kamilla@fedora:~/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. 3.15: Создание релиза

Создаю журнал изменений (рис. 3.16).

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

Рис. 3.16: Создание журнала изменений

Добавляю журнал изменений в индекс (рис. 3.17).

```
kamilla@fedora:~/git-extended$ git add CHANGELOG.md
kamilla@fedora:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 8ccae3] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.17: Добавление журнала изменений в индекс

И затем заливаю релизную ветку в основную ветку (рис. 3.18).

```
kamilla@fedora:~/git-extended$ git flow release finish 1.0.0
Переключились на ветку «master»
Связанная ветка «origin/main» отсутствует в вышестоящем репозитории.
(для исправления запустите «git branch --unset-upstream»)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
Связанная ветка «origin/main» отсутствует в вышестоящем репозитории.
(для исправления запустите «git branch --unset-upstream»)
fatal: нет описания ветки?
```

Рис. 3.18: Релизная ветка

Отправляю данные на github с помощью `git push --all`(рис. 3.19).

```
kamilla@fedora:~/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 533 байта | 533.00 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:kakras/git-extended.git
 05c0023..e7ddd83 develop -> develop
 05c0023..8aa8f12 master -> master
```

Рис. 3.19: Отправка данных

Затем с помощью `git push --tags`(рис. 3.20).

```
kamilla@fedora:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 163 байта | 163.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:kakras/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
```

Рис. 3.20: Отправка данных

И создаю релиз на github, для этого использую утилиты работы с github (рис. 3.21).

```
kamilla@fedora:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
Post "https://api.github.com/graphql": dial tcp 140.82.121.6:443: i/o timeout
```

Рис. 3.21: Создание релиза

Создаю ветку для новой функциональности (рис. 3.22).

```
kamilla@fedora:~/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

Объединяю ветку feature_branch с develop (рис. 3.23).

```
kamilla@fedora:~/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 e7ddd83).

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'
```

Рис. 3.23: Объединение ветки

Создаю релиз с версией 1.2.3 (рис. 3.24).

```
kamilla@fedora:~/git-extended$ git flow release start 1.2.3
Переключились на новую ветку «release/1.2.3»

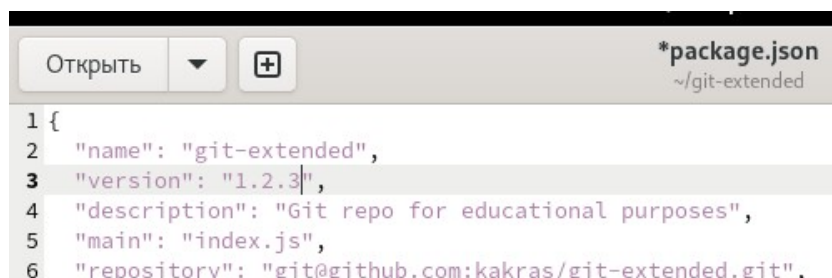
Summary of actions:
- A new branch 'release/1.2.3' was created, based on 'develop'
- You are now on branch 'release/1.2.3'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'
```

Рис. 3.24: Создание релиза

С помощью текстового редактора изменяю файл package.json, обновляя номер версии (рис. 3.25).



```
1 {
2   "name": "git-extended",
3   "version": "1.2.3",
4   "description": "Git repo for educational purposes",
5   "main": "index.js",
6   "repository": "git@github.com:kakras/git-extended.git",
```

Рис. 3.25: Изменение файла

Создаю журнал изменений и добавляю его в индекс (рис. 3.26).


```

kamilla@fedora:~/git-extended$ git standard-changelog
✓ output changes to CHANGELOG.md
kamilla@fedora:~/git-extended$ git add CHANGELOG.md
kamilla@fedora:~/git-extended$ git coomit -am 'chore(site): update changelog'
git: «coomit» не является командой git. Смотрите «git --help».

Самые похожие команды:
  commit
kamilla@fedora:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 9e3882c] chore(site): update changelog
 2 files changed, 5 insertions(+), 1 deletion(-)
kamilla@fedora:~/git-extended$

```

Рис. 3.26: Создание журнала изменений

Заливаю релизную ветку в основную ветку (рис. 3.27).

```

 2 files changed, 5 insertions(+), 1 deletion(-)
kamilla@fedora:~/git-extended$ git flow release finish 1.2.3
Переключились на ветку «master»
Связанная ветка «origin/main» отсутствует в вышестоящем репозитории.
(для исправления запустите «git branch --unset-upstream»)
Merge made by the 'ort' strategy.
 CHANGELOG.md | 4 ++++
 package.json | 2 +-
 2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «master»
Связанная ветка «origin/main» отсутствует в вышестоящем репозитории.
(для исправления запустите «git branch --unset-upstream»)
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 9e3882c).

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'

```

Рис. 3.27: Релизная ветка

Отправляю данные на github с помощью git push --all (рис. 3.28).

```

kamilla@fedora:~/git-extended$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 736 байтов | 736.00 КиБ/с, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:kakras/git-extended.git
   e7ddd83..7cfa11b  develop -> develop
   8aa8f12..4d94dad  master -> master
kamilla@fedora:~/git-extended$

```

Рис. 3.28: Отправка данных

Затем с помощью `git push --tags` (рис. 3.29).

```

kamilla@fedora:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 162 байта | 162.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:kakras/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3
kamilla@fedora:~/git-extended$

```

Рис. 3.29: Отправка данных

Создаю релиз на github с комментариями из журнала изменений (рис. 3.30).

```

kamilla@fedora:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
Post "https://api.github.com/graphql": dial tcp 140.82.121.6:443: i/o

```

Рис. 3.30: Создание релиза

Репозиторий `git-extended` (рис. 3.31).

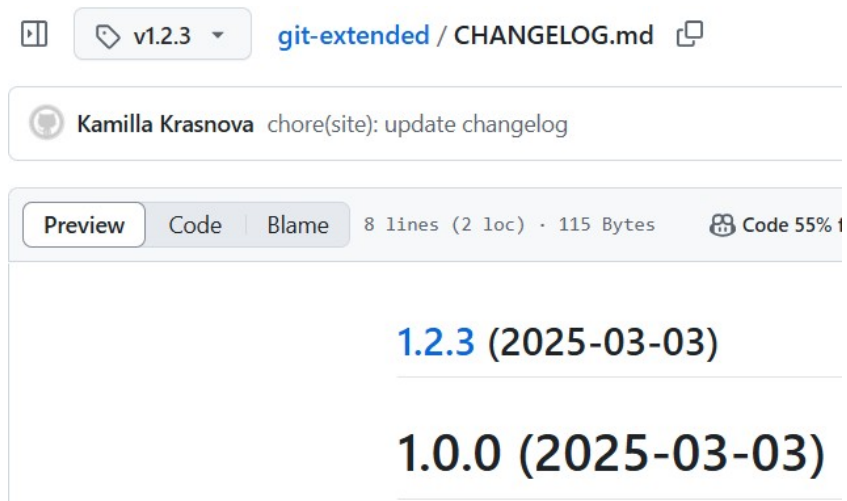


Рис. 3.31: Репозиторий

4 Выводы

В ходе выполнения данной лабораторной работы я получила навыки правильной работы с репозиториями git.

Список литературы