

Отчет по лабораторной работе №4

Продвинутое использование git

Симонова Полина Игоревна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
3.1	Рабочий процесс Gitflow	6
4	Выполнение лабораторной работы	7
4.1	Установка программного обеспечения.	7
4.1.1	Установка git-flow	7
4.1.2	Установка и настройка node.js	7
4.1.3	Общепринятые коммиты.	8
4.2	Практический сценарий использования git.	9
4.2.1	Создание репозитория git.	9
4.2.2	Работа с репозиторием git.	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Установка gitflow в режиме суперпользователя	7
4.2	Установка приложений	8
4.3	Настройка node.js	8
4.4	Настройка программ	8
4.5	Создание репозитория, первый коммит	9
4.6	Настройка пакета	10
4.7	Изменения файла	10
4.8	Выполнение коммита	11
4.9	Команда push	11
4.10	Инициализация gitflow	12
4.11	Установка внешней ветки	12
4.12	Добавление релизной ветки в основную	13
4.13	Команды push -all и push -tags	13
4.14	Создание новой ветки	13
4.15	Объединение веток, создание релиза с более новой версией	14
4.16	Изменение номера версии с 1.0.0 на 1.2.3	14
4.17	Добавление релизной ветки в основную	15
4.18	Отправка данных на гитхаб	15
4.19	Отправка данных на гитхаб	15

1 Цель работы

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 Теоретическое введение

3.1 Рабочий процесс Gitflow

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow. Общая информация Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. 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.

4 Выполнение лабораторной работы

4.1 Установка программного обеспечения.

4.1.1 Установка git-flow

1. Открываем терминал и входим в режим суперпользователя, устанавливаем gitflow. (рис. 4.1).

```
pisimonova@vbox:~$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для pisimonova:
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-in-copr>,
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 репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
pisimonova@vbox:~$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos          3.3 kB/s | 1.5 kB      00:00
Пакет gitflow-1.12.3-1.fc34.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 4.1: Установка gitflow в режиме суперпользователя

4.1.2 Установка и настройка node.js

2. Устанавливаем npm и nodejs. (рис. 4.2).

```

pisimonova@vbox:~$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:00:19 назад, Вт 04 мар
2025 13:40:40.
Пакет nodejs-1:20.18.2-2.fc40.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
pisimonova@vbox:~$ sudo dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:00:35 назад, Вт 04 мар
2025 13:40:40.
Пакет pnpm-9.0.6-1.fc40.noarch уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!

```

Рис. 4.2: Установка приложений

3. Настраиваем nodejs. (рис. 4.3).

```

pisimonova@vbox:~$ pnpm setup
No changes to the environment were made. Everything is already up to date.
pisimonova@vbox:~$ source ~/.bashrc

```

Рис. 4.3: Настройка node.js

4.1.3 Общепринятые коммиты.

4. Настраиваем commitizen и standard-changelog (рис. 4.4).

```

pisimonova@vbox:~$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 6.5s
pisimonova@vbox:~$ source /home/pisimonova/.bashrc
pisimonova@vbox:~$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 5.1s
pisimonova@vbox:~$ pnpm add -g standard-changelog
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 6.7s

```

Рис. 4.4: Настройка программ

4.2 Практический сценарий использования git.

4.2.1 Создание репозитория git.

5. Создаем репозиторий git, настраиваем его и делаем в него первый коммит. (рис. 4.5).

```
pisimonova@vbox:~$ git clone --recursive https://github.com/o5o6am/git-extended.git
Клонирование в «git-extended»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (3/3), готово.
pisimonova@vbox:~$ git commit -m "first commit"
fatal: не найден git репозиторий (или один из его каталогов вплоть до точки монтирования /)
Останавливаю поиск на границе файловой системы (так как GIT_DISCOVERY_ACROSS_FILESYSTEM не установлен).
pisimonova@vbox:~$ cd ~/git-extended
pisimonova@vbox:~/git-extended$ git commit -m "first commit"
Текущая ветка: main
Эта ветка соответствует «origin/main».

ничего коммитить, нет изменений в рабочем каталоге
pisimonova@vbox:~/git-extended$ git remote add origin https://github.com/o5o6am/git-extended.git
error: внешний репозиторий origin уже существует
pisimonova@vbox:~/git-extended$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «https://github.com/o5o6am/git-extended.git»
pisimonova@vbox:~/git-extended$ touch README.md
pisimonova@vbox:~/git-extended$ git add .
pisimonova@vbox:~/git-extended$ git commit -m "first commit"
Текущая ветка: main
Эта ветка соответствует «origin/main».
```

Рис. 4.5: Создание репозитория, первый коммит

6. Настраиваем пакет файлов nodejs (рис. 4.6). В файле package.json меняем необходимые данные. (рис. 4.7).

```

pisimonova@vbox:~/git-extended$ git push -u
branch 'main' set up to track 'origin/main'.
Everything up-to-date
pisimonova@vbox:~/git-extended$ git push
Everything up-to-date
pisimonova@vbox:~/git-extended$ cd ~
pisimonova@vbox:~$ pnpm init
Wrote to /home/pisimonova/package.json

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

```

Рис. 4.6: Настройка пакета

```

{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "https://github.com/o5o6am/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Polina Simonova burntorunn@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 4.7: Изменения файла

7. Выполняем коммит (рис. 4.8). Выкладываем на github. (рис. 4.9).

```

pisimonova@vbox:~/git-extended$ git add .
pisimonova@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):
  (19) change_package.json
? 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
[main ec60fd4] feat: change_package.json
1 file changed, 18 insertions(+)
create mode 100644 package.json

```

Рис. 4.8: Выполнение коммита

```

pisimonova@vbox:~/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.21 КиБ | 1.21 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/o5o6am/git-extended.git
   3ffc843..ec60fd4  main -> main
pisimonova@vbox:~/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/pisimonova/git-extended/.git/hooks]

```

Рис. 4.9: Команда push

8. Инициализируем gitflow, проверяем, на какой ветке мы находимся в данный момент, после чего загружаем весь репозиторий в хранилище. (рис. 4.10).

```

pisimonova@vbox:~/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/pisimonova/git-extended/.git/hooks]
pisimonova@vbox:~/git-extended$ git branch
* develop
main
pisimonova@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/o5o6am/git-extended/pull/new/develop
remote:
To https://github.com/o5o6am/git-extended.git
* [new branch]      develop -> develop

```

Рис. 4.10: Инициализация gitflow

9. Устанавливаем внешнюю ветку как вышестоящую для этой ветки (рис. 4.11).
Создаем релиз с версией 1.0.0 и журнал изменений. (рис. 4.12).

```

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

pisimonova@vbox:~/git-extended$ standard-changelog --first-release

```

Рис. 4.11: Установка внешней ветки

10. Заливаем релизную ветку в основную, добавляем журнал изменений в индекс, после чего заливаем релизную ветку в основную. (рис. 4.13).

```

pisimonova@vbox:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
pisimonova@vbox:~/git-extended$ git add CHANGELOG.md
pisimonova@vbox:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 0bbdb2d] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
pisimonova@vbox:~/git-extended$ git flow release finish 1.0.0

```

Рис. 4.12: Добавление релизной ветки в основную

11. Отправляем данные и теги на гитхаб. (рис. 4.14).

```

pisimonova@vbox:~/git-extended$ git push --all
Перечисление объектов: 2, готово.
Подсчет объектов: 100% (2/2), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (2/2), 1.69 КиБ | 1.69 МБ/с, готово.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/o5o6am/git-extended.git
   ec60fd4..5ded0b3  develop -> develop
pisimonova@vbox:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 164 байта | 164.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/o5o6am/git-extended.git
 * [new tag]         v1.0.0 -> v1.0.0

```

Рис. 4.13: Команды push --all и push --tags

4.2.2 Работа с репозиторием git.

12. Создаем релиз на гитхабе. Создаем ветку для новой функциональности. (рис. 4.15).

```

* [new tag]         v1.0.0 -> v1.0.0
pisimonova@vbox:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/o5o6am/git-extended/releases/tag/v1.0.0
pisimonova@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. 4.14: Создание новой ветки

13. Объединяем новую ветку с develop. Создаём релиз с версией 1.2.3. (рис. 4.16).

```
pisimonova@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 5ded0b3).

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'

pisimonova@vbox:~/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. 4.15: Объединение веток, создание релиза с более новой версией

14. Изменяем номер версии в файле package.json. (рис. 4.17).

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "https://github.com/o5o6am/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Polina Simonova burntorunn@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.16: Изменение номера версии с 1.0.0 на 1.2.3

15. Заливаем релизную ветку в основную (рис. 4.18). Отправляем данные на гитхаб. (рис. 4.19).

```

pisimonova@vbox:~/git-extended$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 ++++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 4 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 2cbbab6a).

```

Рис. 4.17: Добавление релизной ветки в основную

```

pisimonova@vbox:~/git-extended$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.78 КиБ | 1.39 МиБ/с, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/o5o6am/git-extended.git
   5ded0b3..b7458be  develop -> develop
   1c68736..69e8b08  main -> main
pisimonova@vbox:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 166 байтов | 166.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/o5o6am/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3

```

Рис. 4.18: Отправка данных на гитхаб

```

pisimonova@vbox:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/o5o6am/git-extended/releases/tag/v1.2.3
pisimonova@vbox:~/git-extended$

```

Рис. 4.19: Отправка данных на гитхаб

16. Создаём релиз на гитхабе с комментарием из журнала изменений.

5 Выводы

В процессе выполнения лабораторной работы я приобрела навыки правильной работы с репозиториями git.

Список литературы