

Отчёт по лабораторной работе № 4

Продвинутое использование git.

Саенко Ангелина Андреевна

Содержание

Цель работы	6
Задание	7
Выполнение лабораторной работы	8
Выводы	22
Список литературы	23

Список иллюстраций

1	Установка	8
2	Установка	8
3	Установка nodejs	9
4	Установка npm	9
5	Запуск	10
6	Перелогинимся или выполним	10
7	Добавление	10
8	Создание репозитория	11
9	Первый коммит	11
10	Добавление удалённого репозитория в мой локальный Git-проект	11
11	Отправка изменений	12
12	Конфигурация для пакетов Node.js	12
13	Изменяем файл	13
14	Добавление новых файлов	13
15	Коммит	13
16	Выбор	14
17	Отправка	14
18	Инициализация git-flow	14
19	Проверка	15
20	Загрузка репозитория	15
21	Установка внешней ветки	15
22	Создание релиза	15
23	Создание журнала изменений	16
24	Добавление журнала	16
25	Заливание релизной ветки	16
26	Отправка данных	16
27	Отправка данных	17
28	Создание релиза	17
29	Создание ветки	17
30	Объединяем ветки	18
31	Создание релиза	18
32	Обновим номер версии в файле package.json. Установим её в 1.2.3 .	19
33	Создание журнала изменений	19
34	Добавление журнала изменений в индекс	19
35	Зальём ветку	20
36	Отправка данных	20
37	Отправка данных	20

38	Создание релиза	21
----	---------------------------	----

Список таблиц

Цель работы

Получение навыков правильной работы с репозиториями git.

Задание

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

Выполнение лабораторной работы

Для начала установим git-flow из коллекции репозитория (рис. [-@fig:001]).

```
[angelina@asasaenko ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для angelina:
https://copr.fedorainfracloud.org/api_3/rp 100% | 1.4 KiB/s | 652.0 B | 00m00s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является част
ью
основного дистрибутива, и его качество может отличаться.

Проект Fedora не имеет никакой власти над содержанием
за пределами правил, изложенных в Copr FAQ по адресу
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>
и пакеты не имеют никаких требований к качеству или уровню безопасности.

Пожалуйста, не публикуйте сообщения об ошибках, связанных с этими пакетами, в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.
Is this ok [y/N]: y
[angelina@asasaenko ~]$
```

Рис. 1: Установка

```
[angelina@asasaenko ~]$ sudo dnf install gitflow
Обновление и загрузка репозитория:
Copr repo for gitflow owned by elegos 100% | 1.2 KiB/s | 2.4 KiB | 00m
Репозитории загружены.
Пакет Арх. Версия Репозиторий Разм
Установка:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedorainf 261.7 K
Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 57 KiB. Необходимо загрузить 57 KiB.
После этой операции будут использоваться дополнительные 262 KiB (установка 262 KiB,
аление 0 B).
Is this ok [y/N]: y
```

Рис. 2: Установка

На Node.js базируется программное обеспечение для семантического версион-

нирования и общепринятых коммитов. Установим ПО

```
[angelina@saasenko ~]$ sudo dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
Установка зависимостей:
nodejs-libs x86_64 1:22.14.0-2.fc41 updates 75.9 MiB
Установка слабых зависимостей:
nodejs-docs noarch 1:22.14.0-2.fc41 updates 93.7 MiB
nodejs-fulli18n x86_64 1:22.14.0-2.fc41 updates 30.4 MiB
nodejs-npm x86_64 1:10.9.2-1.22.14.0.2.fc41 updates 9.3 MiB

Сводка транзакции:
Установка: 5 пакетов

Общий размер входящих пакетов составляет 39 MiB. Необходимо загрузить 39 MiB.
После этой операции будут использоваться дополнительные 210 MiB (установка 210 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/5] nodejs-1:22.14.0-2.fc41.x86_64 100% | 577.0 KiB/s | 46.7 KiB | 00m00s
[2/5] nodejs-npm-1:10.9.2-1.22.14.0.2.fc41.x86_64 100% | 1.1 MiB/s | 2.4 MiB | 00m00s
[3/5] nodejs-libs-1:22.14.0-2.fc41.x86_64 60% | 88.5 KiB/s | 11.6 MiB | 01m00s
[4/5] nodejs-docs-1:22.14.0-2.fc41.noarch 98% | 127.7 KiB/s | 8.7 MiB | 00m00s
```

Рис. 3: Установка nodejs

```
[angelina@saasenko ~]$ sudo dnf install npm
Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожалуйста, войдите в систему как пользователь с повышенными правами или используйте опции "--as sulepo" или "--downloadonly", чтобы выполнить команду без изменения состояния системы.
[angelina@saasenko ~]$ sudo 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

Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 3 MiB. Необходимо загрузить 3 MiB.
После этой операции будут использоваться дополнительные 16 MiB (установка 16 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] npm-9.13.0-1.fc41.noarch 100% | 4.0 MiB/s | 3.0 MiB | 00m01s
-----
[1/1] Total 100% | 1.1 MiB/s | 3.0 MiB | 00m03s
Выполнение транзакции
[1/3] Проверить файлы пакета 100% | 50.0 B/s | 1.0 B | 00m00s
```

Рис. 4: Установка npm

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

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

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

To start using pnpm, run:
source /home/angelina/.bashrc
```

Рис. 5: Запуск

```
angelina@aaasaenko:~$ source /home/angelina/.bashrc
angelina@aaasaenko:~$ pnpm setup
```

Рис. 6: Перелогинимся или выполним

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

```
angelina@aaasaenko:~$ 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.

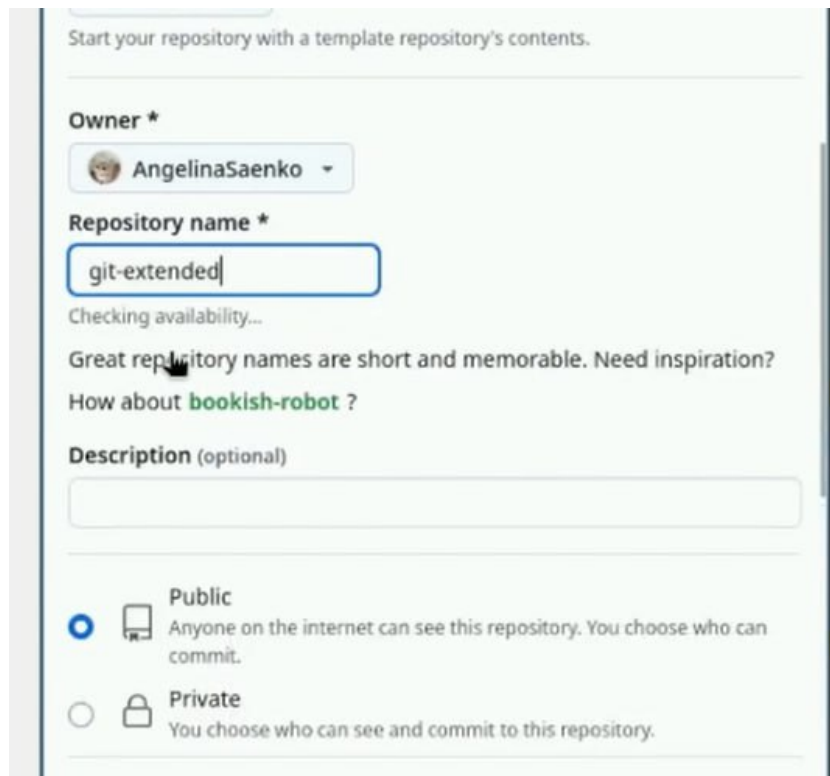
Follow @pnpmjs for updates: https://x.com/pnpmjs

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

/home/angelina/.local/share/pnpm/global/5:
```

Рис. 7: Добавление

Создаём репозиторий на GitHub. Для примера назовём его git-extended



Start your repository with a template repository's contents.

Owner *
AngelinaSaenko

Repository name *
git-extended
Checking availability...

Great repository names are short and memorable. Need inspiration?
How about **bookish-robot** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Рис. 8: Создание репозитория

Делаем первый коммит и выкладываем на github

```
[angelina@aasaenko git-extended]$ git commit -m "first commit"
[main (корневой коммит) b6f1d1b] first commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md
[angelina@aasaenko git-extended]$
```

Рис. 9: Первый коммит

```
[angelina@aasaenko git-extended]$ git remote add origin git@github.com:AngelinaSaenko:git-extended.git
error: внешний репозиторий origin уже существует
[angelina@aasaenko git-extended]$
```

Рис. 10: Добавление удалённого репозитория в мой локальный Git-проект

```
[angelina@asasaenko git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 875 байтов | 875.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AngelinaSaenko/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[angelina@asasaenko git-extended]$
```

Рис. 11: Отправка изменений

Далее сделаем конфигурацию для пакетов Node.js

```
[angelina@asasaenko git-extended]$ npm init
Wrote to /home/angelina/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"
}
[angelina@asasaenko git-extended]$ nano
```

Рис. 12: Конфигурация для пакетов Node.js

Необходимо заполнить несколько параметров пакета. Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования КОММИТОВ

```

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com: AngelinaSaenko/git-extended.git",
  "author": "Angelina Saenko angelinasaenko867@gmail.com",
  "keywords": [],
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 13: Изменяем файл

Добавим новые файлы

```

[angelina@aasaenko git-extended]$ git add .
[angelina@aasaenko git-extended]$ git cz

```

Рис. 14: Добавление новых файлов

Выполним коммит

```

[angelina@aasaenko git-extended]$ git cz
cz v1.0.0 2.1. cz-conventional-changelog@2.3.0

```

Рис. 15: Коммит

Выбираем нужное

```

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) readme.md
? Write a short, imperative tense description of the change (max 83 chars):
  (10) added file
? 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 3f42bbc] feat(readme.md): added file
1 file changed, 15 insertions(+)
create mode 100644 package.json

```

Рис. 16: Выбор

Отправим на github

```

package.json README.md
[angelina@aasaenko git-extended]$ git push
Everything up-to-date
[angelina@aasaenko git-extended]$ git flow init

```

Рис. 17: Отправка

Инициализируем git-flow

```

[angelina@aasaenko 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/angelina/git-extended/.git/hooks]

```

Рис. 18: Инициализация git-flow

Проверим , что мы на ветке develop

```
[angelina@aasaenko git-extended]$ git branch
* develop
  main
[angelina@aasaenko git-extended]$
```

Рис. 19: Проверка

Загрузим весь репозиторий в хранилище

```
[angelina@aasaenko 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/AngelinaSaenko/git-extended/pull/new/develop
remote:
To github.com:AngelinaSaenko/git-extended.git
 * [new-branch]      develop -> develop
```

Рис. 20: Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки

```
[angelina@aasaenko git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 21: Установка внешней ветки

Создадим релиз с версией 1.0.0

```
[angelina@aasaenko 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. 22: Создание релиза

Создадим журнал изменений

```
[angelina@aasaenko git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 23: Создание журнала изменений

Добавим журнал изменений в индекс

```
[angelina@aasaenko git-extended]$ git add CHANGELOG.md
[angelina@aasaenko git-extended]$ git commit -an 'chore(site): add changelog'
```

Рис. 24: Добавление журнала

Зальём релизную ветку в основную ветку

```
[angelina@aasaenko git-extended]$ git_flow release finish 1.0.0
```

Рис. 25: Заливание релизной ветки

Отправим данные на github

```
[angelina@aasaenko 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% (5/5), готово.
Запись объектов: 100% (5/5), 2.81 КиБ | 719.00 КиБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AngelinaSaenko/git-extended.git
 3f42bbc..8efc0d7 develop -> develop
 3f42bbc..e477dd4 main -> main
```

Рис. 26: Отправка данных


```
[angelina@aasaenko 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:AngelinaSaenko/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
```

Рис. 27: Отправка данных

Создадим релиз на github. Для этого будем использовать утилиты работы с github

```
[angelina@aasaenko git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/AngelinaSaenko/git-extended/releases/tag/v1.0.0
```

Рис. 28: Создание релиза

Создадим ветку для новой функциональности

```
[angelina@aasaenko 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. 29: Создание ветки

Далее, продолжаем работу с git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop

```
[angelina@aasaenko 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 8efc0d7).

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'

[angelina@aasaenko git-extended]$
```

Рис. 30: Объединяем ветки

Создадим релиз с версией 1.2.3

```
[angelina@aasaenko 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'

[angelina@aasaenko git-extended]$
```

Рис. 31: Создание релиза

Обновим номер версии в файле package.json. Установим её в 1.2.3

```

{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:AngelinaSaenko/git-extended.git",
  "author": "Angelina Saenko angelinasaenko867@gmail.com",
  "keywords": [],
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 32: Обновим номер версии в файле package.json. Установим её в 1.2.3

Создадим журнал изменений

```

[angelina@aaasaenko git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md

```

Рис. 33: Создание журнала изменений

Добавим журнал изменений в индекс

```

✓ output changes to CHANGELOG.md
[angelina@aaasaenko git-extended]$ git add CHANGELOG.md
[angelina@aaasaenko git-extended]$ git commit -an 'chore(site): update changelog'
[release/1.2.3 decfb84] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[angelina@aaasaenko git-extended]$

```

Рис. 34: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку

```

[angelina@asaenko 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 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 decfb84).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- 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'

[angelina@asaenko git-extended]$

```

Рис. 35: Зальём ветку

Отправим данные на github

```

[angelina@asaenko 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), 2.78 КиБ | 2.78 МБ/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 github.com:AngelinaSaenko/git-extended.git
   8efc0d7..89ddc4f  develop -> develop
   e477dd4..5fe688a  main -> main

```

Рис. 36: Отправка данных

```

[angelina@asaenko git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 163 байта | 81.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AngelinaSaenko/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3

```

Рис. 37: Отправка данных

Создадим релиз на github с комментарием из журнала изменений

```
[angelina@saasaenko git-extended]$ gh release create v1.2.3 -F CHANGELOG.md  
https://github.com/AngelinaSaenko/git-extended/releases/tag/v1.2.3  
✔ Creating release v1.2.3 with 1 commit
```

Рис. 38: Создание релиза

Выводы

Освоены инструменты `git-flow`, `commitizen` и `standard-changelog` для управления ветками, семантического версионирования и оформления коммитов. Настроен репозиторий с поддержкой общепринятых коммитов, созданы и завершены релизы с автоматическим обновлением версий и журналов изменений. Получены навыки эффективного управления разработкой и выпуском проектов.

Список литературы