

Лабораторная работа №4

Отчёт

Ермишина Мария Кирилловна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	17

Список иллюстраций

3.1	Установка git-flow	8
3.2	Установка Node.js	9
3.3	Настройка Node.js	10
3.4	Создание репозитория	11
3.5	Структура файла	12
3.6	Новая ветка	15
3.7	Создание релиза	15
3.8	Обновление инф. в файле	16

Список таблиц

1 Цель работы

Целью данной работы является получение навыков правильной работы с репозиториями git. [tuis4?]

2 Теоретическое введение

1. Gitflow Workflow опубликована и популяризована Винсентом Дриссенем.
2. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта.
3. Данная модель отлично подходит для организации рабочего процесса на основе релизов.
4. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.
5. 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.

3 Выполнение лабораторной работы

1. Установка git-flow Установка из коллекции репозиторияев Copr происходит благодаря следующим командам: (рис. 3.1)

- `sudo dnf copr enable elegos/gitflow`
- `sudo dnf install gitflow`

```
[ermimash@ermimash ~]$ sudo dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rp 100% | 1.5 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 содержимым
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
[ermimash@ermimash ~]$ dnf install gitflow
Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожа
луйста, войдите в систему как пользователь с повышенными правами или используйте опции "
--assumeno" или "--downloadonly", чтобы выполнить команду без изменения состояния си
стемы.
[ermimash@ermimash ~]$ sudo dnf install gitflow
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates 100% | 63.4 KiB/s | 25.6 KiB | 00m00s
Copr repo for gitflow owned by elegos 100% | 3.8 KiB/s | 2.4 KiB | 00m01s
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedora 261.7 KiB

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,
удаление 0 B).
Is this ok [y/N]: y
[1/1] gitflow-0:1.12.3-1.fc34.x86_64 100% | 490.9 KiB/s | 57.4 KiB | 00m00s
-----
57.4 KiB Total 100% | 490.9 KiB/s | 57.4 KiB | 00m00s
```

Рис. 3.1: Установка git-flow

2. Установка Node.js На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Устанавливаем с помощью команд: (рис. 3.2)

- `dnf install nodejs`
- `dnf install pnpm`


```
[ermimash@ermimash ~]$ 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-full-i18n 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% | 213.4 KiB/s | 46.7 KiB | 00m00s
[2/5] nodejs-npm-1:10.9.2-1.22.14.0.2.fc41.x86_64 100% | 5.1 MiB/s | 2.4 MiB | 00m00s
[3/5] nodejs-libs-1:22.14.0-2.fc41.x86_64 100% | 19.1 MiB/s | 19.2 MiB | 00m01s
[4/5] nodejs-docs-1:22.14.0-2.fc41.noarch 100% | 7.3 MiB/s | 8.8 MiB | 00m01s
[5/5] nodejs-full-i18n-1:22.14.0-2.fc41.x86_64 100% | 10.4 MiB/s | 8.5 MiB | 00m01s
-----
[5/5] Total 100% | 20.7 MiB/s | 39.0 MiB | 00m02s
Выполнение транзакции
[1/7] Проверить файлы пакетов 100% | 34.0 B/s | 5.0 B | 00m00s
[2/7] Подготовить транзакцию 100% | 11.0 B/s | 5.0 B | 00m00s
[3/7] Установка nodejs-libs-1:22.14.0-2.fc41.x86_64 100% | 130.2 MiB/s | 75.9 MiB | 00m01s
[4/7] Установка nodejs-1:22.14.0-2.fc41.x86_64 100% | 5.3 MiB/s | 151.6 KiB | 00m00s
[5/7] Установка nodejs-npm-1:10.9.2-1.22.14.0.2.fc41.x86_64 100% | 7.2 MiB/s | 9.8 MiB | 00m01s
[6/7] Установка nodejs-full-i18n-1:22.14.0-2.fc41.x86_64 100% | 149.6 MiB/s | 30.4 MiB | 00m00s
[7/7] Установка nodejs-docs-1:22.14.0-2.fc41.noarch 100% | 8.0 MiB/s | 93.9 MiB | 00m12s
Завершено!
[ermimash@ermimash ~]$ sudo dnf install pnpm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
```

Рис. 3.2: Установка Node.js

3. Настройка Node.js Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. Сначала запускаем, а после перелогиниваемся и выполняем: (рис. 3.3)

- pnpm setup
- source ~/.bashrc

4. Общепринятые коммиты

5. commitizen: (рис. 3.3)

- pnpm add -g commitizen

2. standard-changelog (рис. 3.3)

- `pnpm add -g standard-changelog`

```
[ermimash@ermimash ~]$ pnpm setup
Appended new lines to /home/ermimash/.bashrc

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

To start using pnpm, run:
source /home/ermimash/.bashrc
[ermimash@ermimash ~]$ source ~/.bashrc
[ermimash@ermimash ~]$ pnpm add -g commitizen

Update available! 9.13.0 -> 10.6.1.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.6.1
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/ermimash/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 8.6s
[ermimash@ermimash ~]$ 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/ermimash/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

Done in 4.9s
```

Рис. 3.3: Настройка Node.js

5. Создание репозитория git

6. Подключение репозитория к github Создаём репозиторий на GitHub. Для примера назовём его git-extended. Делаем первый коммит и выкладываем на github: (рис. 3.4)

- `git commit -m "first commit"`
- `git remote add origin (SSH репозитория)`
- `git push -u origin master` (используем `main`, так как не задавали название ветки)

```
[ermimash@ermimash git-extended]$ git commit -m "first commit"
[main (корневой коммит) ac5e6e1] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
[ermimash@ermimash git-extended]$ git push -u origin maste
error: src refspec maste ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:ErmiMash/git-extended.git»
[ermimash@ermimash git-extended]$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:ErmiMash/git-extended.git»
[ermimash@ermimash git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 873 байта | 873.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ErmiMash/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.4: Создание репозитория

2. Конфигурация общепринятых коммитов
Конфигурация для пакетов Node.js выполняется с помощью команды `npm init`. (рис. ??) В файле необходимо изменить данные, приводим файл к следующему виду: (рис. 3.5)

```
{ "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" } } }
```

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:ErmiMash/git-extended.git",
  "author": "Mariya Ermishina 1132230166@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.5: Структура файла

Далее отправляем файлы на Git: (рис. ??), (рис. ??) - `git add .` (добавим новые файлы) - `git cz` (выполним коммит) - `git push` (отправим на github)

```
[ermimash@ermimash git-extended]$ npm init
Wrote to /home/ermimash/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"
}
[ermimash@ermimash git-extended]$ nano package.json
```

```
[ermimash@ermimash 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
? What is the scope of this change (e.g. component or feature)?
(skip) readme.md
? Write a short, imperative tense description of the change:
(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 6aa6a7d] feat(readme.md): added file
1 file changed, 14 insertions(+)
create mode 100644 package.json
[ermimash@ermimash git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.15 КиБ | 591.00 КиБ/с,
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:ErmiMash/git-extended.git
ac5e6e1..6aa6a7d main -> main
```

3. Конфигурация git-flow Инициализируем git-flow с помощью `git flow init`. Префикс для ярлыков установим в `v`. После проверяем нахождение на ветке `develop`: (рис. ??)

- `git branch`

Загрузите весь репозиторий в хранилище с помощью `git push --all`. (рис. ??)

Установите внешнюю ветку как вышестоящую для этой ветки: (рис. ??) - `git branch --set-upstream-to=origin/develop develop`

Создадим релиз с версией 1.0.0: (рис. ??) - `git flow release start 1.0.0`

Создадим журнал изменений: (рис. ??) - `standard-changelog --first-release`

Добавим журнал изменений в индекс: (рис. ??) - `git add CHANGELOG.md - git commit -am 'chore(site): add changelog'`

Зальём релизную ветку в основную ветку: (рис. ??) - `git flow release finish 1.0.0`

Отправим данные на github: (рис. ??) - `git push --all - git push --tags`

Создадим релиз на github. Для этого будем использовать утилиты работы с github: (рис. ??) - `gh release create v1.0.0 -F CHANGELOG.md`

```
[ermimash@ermimash git-extended]$ git flow init -f
Which branch should be used for bringing forth production releases?
  - develop
  - main
Branch name for production releases: [main]
Which branch should be used for integration of the "next release"?
  - develop
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/ermimash/git-extended/.git/hooks]
[ermimash@ermimash git-extended]$
[ermimash@ermimash git-extended]$ git branch
* develop
  main
```

```
[ermimash@ermimash git-extended]$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub
remote:   https://github.com/ErmiMash/git-extended
remote:
To github.com:ErmiMash/git-extended.git
 * [new branch]      develop -> develop
[ermimash@ermimash git-extended]$ git branch --set-up
p
branch 'develop' set up to track 'origin/develop'.
[ermimash@ermimash git-extended]$ git flow release st
Переключились на новую ветку «release/1.0.0»

Summary of actions:
- A new branch 'release/1.0.0' was created, based on
- You are now on branch 'release/1.0.0'

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

    git flow release finish '1.0.0'

[ermimash@ermimash git-extended]$ standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[ermimash@ermimash git-extended]$ git add CHANGELOG.m
[ermimash@ermimash git-extended]$ git commit -am 'cho
[release/1.0.0 85d5ab3] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```



```

[ermimash@ermimash git-extended]$ git flow release finish 1.0.0
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 9 ++++++++
  1 file changed, 9 insertions(+)
  create mode 100644 CHANGELOG.md
Уже на «main»
Ваша ветка опережает «origin/main» на 2 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 | 9 ++++++++
  1 file changed, 9 insertions(+)
  create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 85d5ab3).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- 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'

[ermimash@ermimash git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.77 КиБ | 2.77 МБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ErmiMash/git-extended.git
   6aa6a7d..d22b26f develop -> develop
   6aa6a7d..4699074 main -> main
[ermimash@ermimash git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байтов | 155.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ErmiMash/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
[ermimash@ermimash git-extended]$ gh release create v1.0.0
https://github.com/ErmiMash/git-extended/releases/tag/v1.0.0

```

6. Работа с репозиторием git

7. Разработка новой функциональности (рис. 3.6) Создадим ветку для новой функциональности:

- git flow feature start feature_branch

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

```
[ermimash@ermimash 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

[ermimash@ermimash 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 d22b26f).

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. 3.6: Новая ветка

2. Создание релиза git-flow Создадим релиз с версией 1.2.3: (рис. 3.7)

- git flow release start 1.2.3

```
[ermimash@ermimash 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'

[ermimash@ermimash git-extended]$ nano package.json
```

Рис. 3.7: Создание релиза

Обновляем номер версии в файле package.json. Установите её в 1.2.3. (рис. 3.8)

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:ErmiMash/git-extended.git",
  "author": "Mariya Ermishina 1132230166@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.8: Обновление инф. в файле

Создадим журнал изменений: (рис. ??) - standard-changelog

Добавим журнал изменений в индекс: (рис. ??) - git add CHANGELOG.md - git commit -am 'chore(site): update changelog'

Зальём релизную ветку в основную ветку: (рис. ??) - git flow release finish 1.2.3

Отправим данные на github: (рис. ??) - git push --all - git push --tags

Создадим релиз на github с комментарием из журнала изменений: (рис. ??) - gh release create v1.2.3 -F CHANGELOG.md

```
[ermimash@ermimash git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[ermimash@ermimash git-extended]$ git add CHANGELOG.md
[ermimash@ermimash git-extended]$ git commit -am 'chore(site): update changelog'
[release/1.2.3 646651b] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[ermimash@ermimash 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 646651b).

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'
```

```
[ermimash@ermimash git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.75 КиБ | 2.75 МБ/с, г
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with
To github.com:ErmiMash/git-extended.git
d22b26f..754330a develop -> develop
4699074..a6b676d main -> main
[ermimash@ermimash git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 157 байтов | 157.00 КиБ/с
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:ErmiMash/git-extended.git
* [new tag] v1.2.3 -> v1.2.3
[ermimash@ermimash git-extended]$ gh release create v
https://github.com/ErmiMash/git-extended/releases/tag
[ermimash@ermimash git-extended]$
```


4 Выводы

В ходе данной работы были получены навыки правильной работы с репозиториями git.