

Отчет по выполнению лабораторной работы №4

Операционные системы

Намруев Максим Саналович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Установка git-flow	7
3.2	Установка git-flow	8
3.3	Установка node.js	8
3.4	Запуск npnm	9
3.5	Выполнение source	9
3.6	Выполнение commitizen	9
3.7	Выполнение standart-changelog	9
3.8	Создание нового репозитория	10
3.9	Клонирование репозитория	10
3.10	Первый коммит	10
3.11	Конфигурация пакетов	11
3.12	Заполнение параметров пакета	11
3.13	Добавление файлов и выполнение коммита	11
3.14	Отправка файлов на github	12
3.15	Инициализация git-flow	12
3.16	Проверка ветки	12
3.17	Загрузка репозитория в хранилище	12
3.18	Установка внешней ветки	13
3.19	Установка релиза 1.0.0	13
3.20	Создание журнала изменений	13
3.21	Добавление индекса в журнал изменений	13
3.22	Залив релизной ветки в основную	14
3.23	Отправка файлов на гитхаб	14
3.24	Отправка данных на гитхаб	14
3.25	Создание релиза на github	14
3.26	Создание новой ветки	15
3.27	объединение веток	15
3.28	Создание релиза 1.2.3	15
3.29	Обновление версии	16
3.30	Создание журнала изменений	16
3.31	Добавление журнала изменений в индекс	16
3.32	Добавление релизной ветки в основную	17
3.33	Отправка данных на github	17
3.34	Отправка данных на github	17
3.35	Создание релиза на github	17

Список таблиц

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 Выполнение лабораторной работы

Устанавливаю git-flow из коллекции репозитория Copr (рис. 3.1). (рис. 3.2).

```
[msnamruev@msnamruev ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для msnamruev:
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 репозит
ия.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
[msnamruev@msnamruev ~]$ dnf install gitflow
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на больш
стве систем – под именем пользователя root).
[msnamruev@msnamruev ~]$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos          3.1 kB/s | 1.5 kB      00:00
Зависимости разрешены.
```

Рис. 3.1: Установка git-flow

```

[msnamruev@msnamruev ~]$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos          3.1 kB/s | 1.5 kB    00:00
Зависимости разрешены.
=====
Пакет  Архитектура
      Версия      Репозиторий      Размер
=====
Установка:
  gitflow
      x86_64 1.12.3-1.fc34 copr:copr.fedorainfracloud.org:elegos:gitflow 57 k
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/Н]: y
Загрузка пакетов:
gitflow-1.12.3-1.fc34.x86_64.rpm          311 kB/s | 57 kB    00:00
=====
Общий размер          306 kB/s | 57 kB    00:00
Copr repo for gitflow owned by elegos     13 kB/s | 998 B    00:00
Импорт GPG-ключа 0x80F63AA3:

```

Рис. 3.2: Установка git-flow

Устанавливаю Node.js (рис. 3.3).

```

[msnamruev@msnamruev ~]$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:00:23 назад, Сб 09 мар
2024 14:40:22.
Зависимости разрешены.
=====
Пакет      Архитектура
          Версия      Репозиторий      Размер
=====
Установка:
  nodejs    x86_64 1:18.19.0-1.fc38 updates          2.1 М
Установка зависимостей:
  nodejs-libs x86_64 1:18.19.0-1.fc38 updates          14 М
Установка слабых зависимостей:
  nodejs-docs noarch 1:18.19.0-1.fc38 updates          7.9 М
  nodejs-full-i18n x86_64 1:18.19.0-1.fc38 updates          8.5 М
  nodejs-npm x86_64 1:10.2.3-1.18.19.0.1.fc38 updates          2.2 М
=====
Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 35 М
Объем изменений: 185 М

```

Рис. 3.3: Установка node.js

Запускаю npm.(рис. 3.4).


```

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

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

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

```

Рис. 3.4: Запуск pnpm

Перелогиниваюсь и выполняю source ~/.bashrc (рис. 3.5).

```

[msnamruev@msnamruev ~]$ source /home/msnamruev/.bashrc

```

Рис. 3.5: Выполнение source

Выполняю коммит commitizen.(рис. 3.6).

```

[msnamruev@msnamruev ~]$ pnpm add -g commitizen
Packages: +152
+-----+
Downloading registry.npmjs.org/typescript/5.4.2: 5,82 MB/5,82 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/home/msnamruev/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 6.8s
[msnamruev@msnamruev ~]$

```

Рис. 3.6: Выполнение commitizen

Выполняю коммит standart-changelog (рис. 3.7).

```

Done in 6.8s
[msnamruev@msnamruev ~]$ pnpm add -g standart-changelog
Packages: +56
+-----+
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/home/msnamruev/.local/share/pnpm/global/5:
+ standart-changelog 5.0.0

Done in 4.8s
[msnamruev@msnamruev ~]$

```

Рис. 3.7: Выполнение standart-changelog

Создаю новый репозиторий git-extended.(рис. 3.8).

Owner * msnamruev / Repository name * git-extended

✔ git-extended is available.

Great repository names are short and memorable. Need inspiration? How about [solid-oc](#)

Description (optional)

Рис. 3.8: Создание нового репозитория

Перехожу в папку work и клонирую его.(рис. 3.9).

```
[msnamruev@msnamruev ~]$ cd work
[msnamruev@msnamruev work]$ git clone --recursive git@github.com:msnamruev/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
[msnamruev@msnamruev work]$ ls
```

Рис. 3.9: Клонирование репозитория

Перехожу в него и делаю первый коммит.(рис. 3.10).

```
[msnamruev@msnamruev git-extended]$ echo "# git-extended" >> README.md
[msnamruev@msnamruev git-extended]$ git init
Переинициализирован существующий репозиторий Git в /home/msnamruev/work/git-extended/.git/
[msnamruev@msnamruev git-extended]$ git add README.md
[msnamruev@msnamruev git-extended]$ git commit -m "first commit"
main (корневой коммит) 26dfba8] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
[msnamruev@msnamruev git-extended]$ git branch -M main
[msnamruev@msnamruev git-extended]$ git remote add origin git@github.com:msnamruev/git-extended
.git
error: внешний репозиторий origin уже существует
[msnamruev@msnamruev git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 882 байта | 882.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
to github.com:msnamruev/git-extended.git
* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.10: Первый коммит

Далее выполняю конфигурацию для пакетов node.js.(рис. 3.11).

```
[msnamruev@msnamruev git-extended]$ pnpm init
Wrote to /home/msnamruev/work/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 пакетов

Далее заполняю несколько параметров пакета.(рис. 3.12).

```
package.json  [-M--]  9 L:[ 1+12 13/ 15] *(382 / 385b) 0010 0x00A
{
  "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: Заполнение параметров пакета

Добавляю новые файлы и выполняю коммит.(рис. 3.13).

```
[msnamruev@msnamruev git-extended]$ git add .
[msnamruev@msnamruev git-extended]$ git cz
cz-cli@4.3.0, 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):
(3) idk
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? Yes
? A BREAKING CHANGE commit requires a body. Please enter a longer description
itself:
change files
? Describe the breaking changes:
```

Рис. 3.13: Добавление файлов и выполнение коммита

Далее отправляю файлы на github.(рис. 3.14).

```
[msnamruev@msnamruev git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.16 КиБ | 1.16 МБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:msnamruev/git-extended.git
 26dfba8..9eeef27  main -> main
[msnamruev@msnamruev git-extended]$
```

Рис. 3.14: Отправка файлов на github

Инициализируем git-flow.(рис. 3.15).

```
[msnamruev@msnamruev 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/msnamruev/work/git-extended/.git/hooks]
[msnamruev@msnamruev git-extended]$
```

Рис. 3.15: Инициализация git-flow

Проверяю, что нахожусь в ветке develop.(рис. 3.16).

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

Рис. 3.16: Проверка ветки

Загружаю весь репозиторий в хранилище.(рис. 3.17).

```
[msnamruev@msnamruev git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/msnamruev/git-extended/pull/new/develop
remote:
To github.com:msnamruev/git-extended.git
 * [new branch]      develop -> develop
[msnamruev@msnamruev git-extended]$
```

Рис. 3.17: Загрузка репозитория в хранилище

Устанавливаю внешнюю ветку как вышестоящую для этой ветки.(рис. 3.18).

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

Рис. 3.18: Установка внешней ветки

Создаю релиз с версией 1.0.0 (рис. 3.19).

```
[msnamruev@msnamruev 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'

[msnamruev@msnamruev git-extended]$
```

Рис. 3.19: Установка релиза 1.0.0

Создаю журнал изменений.(рис. 3.20).

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

Рис. 3.20: Создание журнала изменений

Добавляю в журнал изменений индекс.(рис. 3.21).

```
[msnamruev@msnamruev git-extended]$ git add CHANGELOG.md
[msnamruev@msnamruev git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 46966e2] chore(site): add changelog
1 file changed, 14 insertions(+)
create mode 100644 CHANGELOG.md
[msnamruev@msnamruev git-extended]$
```

Рис. 3.21: Добавление индекса в журнал изменений

Заливаю релизную ветку в основную ветку.(рис. 3.22).

```
[msnamruev@msnamruev 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 | 14 ++++++++
  1 file changed, 14 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)
подсказка: Ожидание, пока вы закроете редактор с файлом...

Используйте «fg» для возврата в nano

[1]+  Остановлен   git flow release finish 1.0.0
[msnamruev@msnamruev git-extended]$
```

Рис. 3.22: Залив релизной ветки в основную

Отправляю данные на гитхаб.(рис. 3.23). (рис. 3.24).

```
[msnamruev@msnamruev git-extended]$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.97 КиБ | 1.97 МБ/с, готово.
Всего 4 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:msnamruev/git-extended.git
  9eeef27..a6e71f8  main -> main
* [new branch]      release/1.0.0 -> release/1.0.0
[msnamruev@msnamruev git-extended]$
```

Рис. 3.23: Отправка файлов на гитхаб

```
[msnamruev@msnamruev git-extended]$ git push --tags
Everything up-to-date
[msnamruev@msnamruev git-extended]$
```

Рис. 3.24: Отправка данных на гитхаб

Создаю релиз на github (рис. 3.25).

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

Рис. 3.25: Создание релиза на github

Создаю ветку для новой функциональности (рис. 3.26).

```
[msnamruev@msnamruev 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

[msnamruev@msnamruev git-extended]$
```

Рис. 3.26: Создание новой ветки

Объединяю ветку feature_branch с develop.(рис. 3.27).

```
[msnamruev@msnamruev 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 9eeef27).

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'

[msnamruev@msnamruev git-extended]$
```

Рис. 3.27: объединение веток

Создаю релиз с версией 1.2.3 (рис. 3.28).

```
[msnamruev@msnamruev 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. 3.28: Создание релиза 1.2.3

Обновляю номер версии в файле package.json.(рис. 3.29).

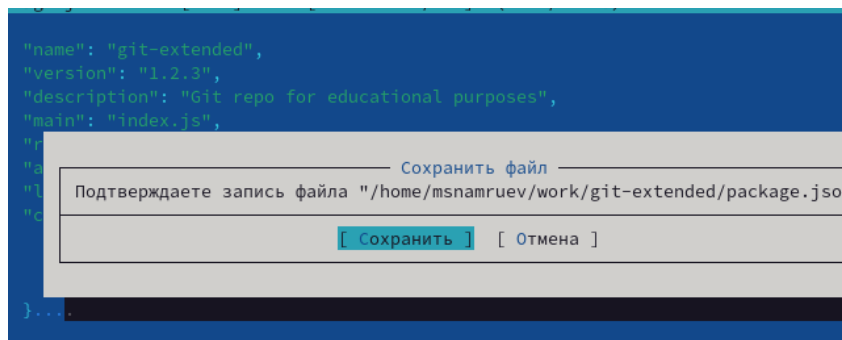


Рис. 3.29: Обновление версии

Создаю журнал изменений.(рис. 3.30).

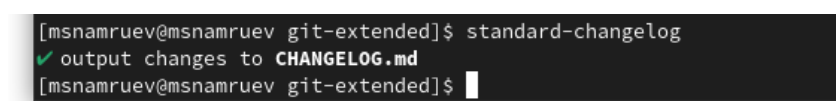


Рис. 3.30: Создание журнала изменений

Добавляю журнал изменений в индекс.(рис. 3.31).

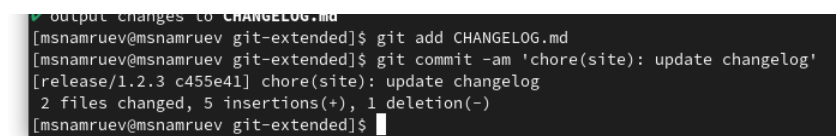


Рис. 3.31: Добавление журнала изменений в индекс

Заливаю релизную ветку в основную ветку.(рис. 3.32).


```

2 files changed, 5 insertions(+), 1 deletion(-)
[msnamruev@msnamruev 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 c455e41).

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

```

Рис. 3.32: Добавление релизной ветки в основную

Отправляю данные на github.(рис. 3.33).(рис. 3.34).

```

[msnamruev@msnamruev git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.78 КиБ | 2.78 МБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:msnamruev/git-extended.git
   969fb30..85e284a  develop -> develop
   ef9e444..8b2e0ad  main -> main
[msnamruev@msnamruev git-extended]$

```

Рис. 3.33: Отправка данных на github

```

[msnamruev@msnamruev 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.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:msnamruev/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3
[msnamruev@msnamruev git-extended]$

```

Рис. 3.34: Отправка данных на github

Создаю релиз на github с комментарием из журнала изменений.(рис. 3.35).

```

[msnamruev@msnamruev git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/msnamruev/git-extended/releases/tag/v1.2.3
[msnamruev@msnamruev git-extended]$

```

Рис. 3.35: Создание релиза на github

4 Выводы

После выполнения данной лабораторной работы я приобрел навыки правильной работы с репозиториями git.

Список литературы