

Отчет по лабораторной работе 4

Продвинутое использование git

Неустроева Ирина Николаевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Добавление общепринятых коммитов.	8
3.3	Конфигурация общепринятых коммитов.	9
3.4	Конфигурация git-flow.	11
3.5	Разработка новой функциональности.	13
3.6	Создание релиза.	14
4	Вывод	16

Список иллюстраций

3.1	Установка git-flow	7
3.2	Установка Node.js.	7
3.3	Добавление каталога	8
3.4	Добавление двух программ	8
3.5	Новый репозиторий	9
3.6	Отправка коммита на гитхаб	9
3.7	Ввод команды	10
3.8	отредактированный файл package.json	10
3.9	Использование скрипта cz для коммита	11
3.10	Инициализация git-flow	12
3.11	Отправка изменений на гитхаб	12
3.12	Создание нового релиза	12
3.13	Настраиваем релиз	13
3.14	ссылка	13
3.15	Создание ветки	13
3.16	объединение веток	14
3.17	Создание ветки	14
3.18	Добавление данных в индекс	15
3.19	Добавление данных на гитхаб	15
3.20	создание на гитхабе новой версии	15

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

3.1 Установка программного обеспечения

Устанавливаем git-flow (рис. 3.1).

```
inneustroeva@inneustroeva:~$ sudo -i
root@inneustroeva:~# dnf copr enable elegos/gitflow
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.
root@inneustroeva:~# dnf install gitflow
Copr repo for gitflow owned by elegos
Fedora 39 - x86_64 - Updates [ ===
```

Рис. 3.1: Установка git-flow

Устанавливаем Node.js. На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. (рис. 3.2).

```
Выполнено!
root@inneustroeva:~# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:44 назад, Пт 08 мар 2024 21:48:29.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия
=====
Установка:
nodejs                x86_64                1:20.10.0-3.fc39
Установка зависимостей:
nodejs-libs            x86_64                1:20.10.0-3.fc39
```

Рис. 3.2: Установка Node.js.

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. (рис. 3.3).

```
root@inneustroeva:~# pnpm setup
Appended new lines to /root/.bashrc

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

Рис. 3.3: Добавление каталога

3.2 Добавление общепринятых коммитов.

При помощи утилиты pnpm добавляем две программы: standard changelog и commitizen. (рис. 3.4).

```
root@inneustroeva:~# pnpm add -g commitizen

Update available! 8.12.0 -> 8.15.4.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v8.15.4
Run "pnpm add -g pnpm" to update.

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

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

/root/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 10.2s
root@inneustroeva:~# pnpm add -g standard-changelog
Progress: resolved 154, reused 151, downloaded 3, added 0
```

Рис. 3.4: Добавление двух программ

Создали новый репозиторий на GitHub (рис. 3.5).

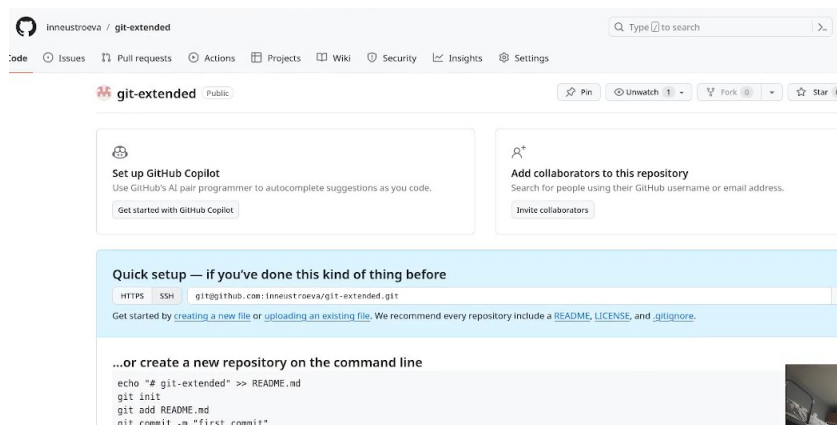


Рис. 3.5: Новый репозиторий

Делаем первый коммит и выкладываем на github (рис. 3.6).

```
inneustroeva@inneustroeva:~/git-extended$ git commit -m "first commit"
[main (корневой коммит) 10e1bcc] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
inneustroeva@inneustroeva:~/git-extended$ git remote add origin git@github.com:inneustroeva/git-extended.git
error: внешний репозиторий origin уже существует
inneustroeva@inneustroeva:~/git-extended$ git remote add new-origin git@github.com:inneustroeva/git-extended.git
inneustroeva@inneustroeva:~/git-extended$ git push -u origin master
```

Рис. 3.6: Отправка коммита на гитхаб

3.3 Конфигурация общепринятых коммитов.

Введем команду `rpm init`. Вывод команды демонстрирует содержание файла и его место в системе. (рис. 3.7).

```

branch main set up to track new-origin/main :
inneustroeva@inneustroeva:~/git-extended$ pnpm init
Wrote to /home/inneustroeva/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. 3.7: Ввод команды

Откроем файл, адрес которого видим в выводе команды, и отредактируем его так, как указано в задании лабораторной работы (рис. 3.8).

```

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes ",
  "main": "index.js",
  "repository": "git@github.com:inneustroeva/git-extended.git",
  "author": "Irina Neustroeva 1132231833@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 3.8: отредактированный файл package.json

После редактирования файла необходимо добавить файлы в репозиторий и выполнить коммит с помощью установленного ранее скрипта. (рис. 3.9).

```
inneustroeva@inneustroeva:~/git-extended$ git cz
No files added to staging! Did you forget to run git add?
inneustroeva@inneustroeva:~/git-extended$ git add .
inneustroeva@inneustroeva:~/git-extended$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: fix:      A bug fix
? 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 84 chars):
  (8) add fail
? 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 e9fc25a] fix(readme.md): add fail
1 file changed, 14 insertions(+)
create mode 100644 package.json
inneustroeva@inneustroeva:~/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 518 байтов | 518.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использо
To github.com:inneustroeva/git-extended.git
10e1bcc..e9fc25a  main -> main
```

Рис. 3.9: Использование скрипта cz для коммита

Коммит успешно настроен и отправлен.

3.4 Конфигурация git-flow.

Инициализируем git-flow и проверим, что мы находимся на нужной ветке (develop) (рис. 3.10).

```
inneustroeva@inneustroeva:~/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/inneustroeva/git-extended/.git/hooks]
inneustroeva@inneustroeva:~/git-extended$ git branch
* develop
main
```

Рис. 3.10: Инициализация git-flow

Отправляем изменения на гитхаб. Следующее, что нужно сделать - установить внешнюю ветку как вышестоящую для этой ветки.(рис. 3.11).

```
inneustroeva@inneustroeva:~/git-extended$ git push --all
Everything up-to-date
inneustroeva@inneustroeva:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
inneustroeva@inneustroeva:~/git-extended$
```

Рис. 3.11: Отправка изменений на гитхаб

После того, как работа с ветками закончена, создаем новый релиз: 1.0.0 (рис. 3.12).

```
inneustroeva@inneustroeva:~/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. 3.12: Создание нового релиза

Далее настраиваем релиз: создаем журнал изменений, добавляем его в индекс и заливаем релиз в основную ветку (рис. 3.13)

```
inneustroeva@inneustroeva:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
inneustroeva@inneustroeva:~/git-extended$ git add CHANGELOG.md
inneustroeva@inneustroeva:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 eed355f] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
inneustroeva@inneustroeva:~/git-extended$ git flow release finish 1.0.0
```

Рис. 3.13: Настраиваем релиз

После того, как отправили данные, создаем релиз на гитхаб и получаем ссылку на него (рис. 3.14)

```
inneustroeva@inneustroeva:~/git-extended$ gh release create 1.0.0 --title "1.0.0" --notes "1.0.0"
https://github.com/inneustroeva/git-extended/releases/tag/1.0.0
```

Рис. 3.14: ссылка

3.5 Разработка новой функциональности.

Создадим ветку для новой функциональности:(рис. 3.15)

```
inneustroeva@inneustroeva:~/git-extended$ git flow feature start feature_branch
A
CHANGELOG.md
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. 3.15: Создание ветки

Далее работа с гит продолжается как обычно. Для тестового репозитория никакую новую функциональность разрабатывать не будем, переходим сразу к следующему шагу: объединение веток feature_branch и develop:(рис. 3.16)

```
inneustroeva@inneustroeva:~/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 e761283).

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'

inneustroeva@inneustroeva:~/git-extended$
```

Рис. 3.16: объединение веток

3.6 Создание релиза.

Создадим релиз с версией 1.2.3 и оказываемся на ветке release/1.2.3: (рис. 3.17)

```
inneustroeva@inneustroeva:~/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
```

Рис. 3.17: Создание ветки

В файле package.json обновляем номер версии. После обновляем журнал изменений и добавляем его на гитхаб: (рис. 3.18)

```

inneustroeva@inneustroeva:~/git-extended$ mc
inneustroeva@inneustroeva:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
inneustroeva@inneustroeva:~/git-extended$ git add CHANGELOG.m
d
git commit -am 'chore(site): update changelog'
[release/1.2.3 cccace5] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
inneustroeva@inneustroeva:~/git-extended$

```

Рис. 3.18: Добавление данных в индекс

Заливаем релиз в основную ветку, после чего отправляем данные на гитхаб:
(рис. 3.19)

```

inneustroeva@inneustroeva:~/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 | 7 +-----
package.json | 2 +-
2 files changed, 2 insertions(+), 7 deletions(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 5 коммитов.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.

```

Рис. 3.19: Добавление данных на гитхаб

Последним шагом закроем релиз и создадим на гитхаб новую версию - 1.2.3:
(рис. 3.20)

```

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

```

Рис. 3.20: создание на гитхабе новой версии

4 Вывод

Получены навыки правильной работы с репозиториями git, выполнена работа для тестового репозитория и дальнейшие преобразования для основного репозитория курса.