

Отчёт по лабораторной работе №4

Продвинутое использование git

Грачева Мария Валерьевна

Содержание

1	Цель работы	1
2	Теоретическое введение	1
3	Выполнение лабораторной работы	1
4	Выводы.....	5
	Список литературы.....	5

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Теоретическое введение

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.

3 Выполнение лабораторной работы

Установка Node.js На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов.

Fedora

```
dnf install nodejs apt-get install pnpm
```

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

Запустите:

```
pnpm setup
```

 Перелогиньтесь, или выполните:

```
source ~/.bashrc
```

Общепринятые коммиты commitizen

Данная программа используется для помощи в форматировании коммитов.

```
pnpm add -g commitizen
```

 При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов. standard-changelog

Данная программа используется для помощи в создании логов.

```
pnpm add -g standard-changelog
```

 Практический сценарий использования git

Создание репозитория git

Подключение репозитория к github

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

Делаем первый коммит и выкладываем на github:

```
git commit -m "first commit" git remote add origin git@github.com:/git-extended.git git push -u origin master
```

 Конфигурация общепринятых коммитов

Конфигурация для пакетов Node.js

```
pnpm init
```

 Необходимо заполнить несколько параметров пакета.

Название пакета. Лицензия пакета. Список лицензий для npm:

<https://spdx.org/licenses/>. Предлагается выбирать лицензию CC-BY-4.0. Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов:

```
"config": { "commitizen": { "path": "cz-conventional-changelog" } }
```

 Таким образом, файл package.json приобретает вид:

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

} Добавим новые файлы:

git add . Выполним коммит:

git cz Отправим на github:

git push Конфигурация git-flow

Инициализируем git-flow

git flow init Префикс для ярлыков установим в v.

Проверьте, что Вы на ветке develop:

git branch Загрузите весь репозиторий в хранилище:

git push --all Установите внешнюю ветку как вышестоящую для этой ветки:

git branch --set-upstream-to=origin/develop develop Создадим релиз с версией 1.0.0

git flow release start 1.0.0 Создадим журнал изменений

standard-changelog --first-release Добавим журнал изменений в индекс

git add CHANGELOG.md git commit -am 'chore(site): add changelog' Зальём релизную ветку в основную ветку

git flow release finish 1.0.0 Отправим данные на github

git push --all git push --tags Создадим релиз на github. Для этого будем использовать утилиты работы с github:

gh release create v1.0.0 -F CHANGELOG.md Работа с репозиторием git

Разработка новой функциональности

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 Создание релиза git-flow

Создадим релиз с версией 1.2.3:

git flow release start 1.2.3 Обновите номер версии в файле package.json. Установите её в 1.2.3. Создадим журнал изменений

standard-changelog Добавим журнал изменений в индекс

git add CHANGELOG.md git commit -am 'chore(site): update changelog' Зальём релизную ветку в основную ветку

git flow release finish 1.2.3 Отправим данные на github

git push --all git push --tags Создадим релиз на github с комментарием из журнала изменений:

gh release create v1.2.3 -F CHANGELOG.md Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 1).

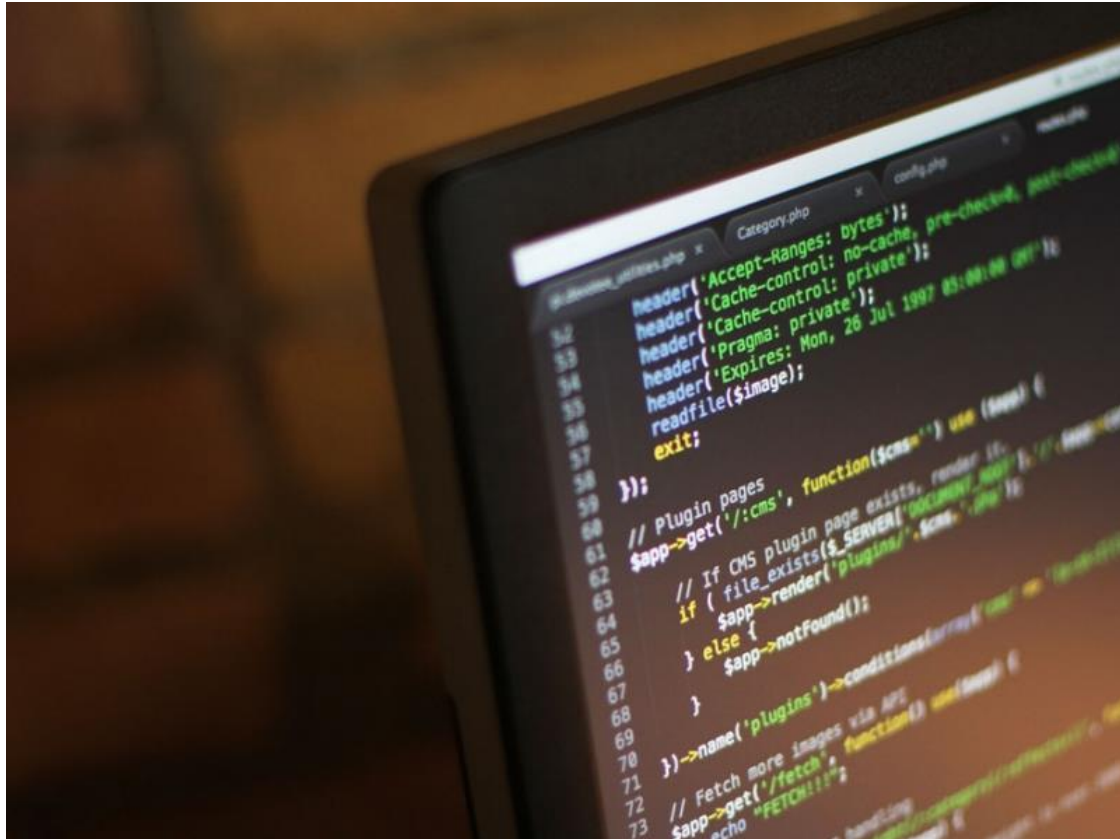


Рис. 1: Пример

(рис. fig. 2).

```
[mvgracheva@fedora ~]$ sudo su
[sudo] пароль для mvgracheva:
'[root@fedora mvgracheva]# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:15:49 назад, Сб 02 мар
2024 18:50:10.
Зависимости разрешены.
=====
Пакет                Архитектура
                        Версия
                        Репозиторий
                        Размер
=====
Установка:
nodejs                x86_64      1:16.18.1-1.fc36      updates      104 k
Установка зависимостей:
nodejs-libs           x86_64      1:16.18.1-1.fc36      updates      13 M
openssl               x86_64      1:3.0.8-1.fc36        updates      1.1 M
```

Рис. 2: Шаг 1

(рис. fig. 3).

```
[root@fedora mvgracheva]# apt-get install pnpm
bash: apt-get: команда не найдена...
Установить пакет «apt», предоставляющий команду «apt-get»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
apt-2.5.6-1.fc36.x86_64      Command-line package manager for Debian packages
apt-libs-2.5.6-1.fc36.x86_64 Runtime libraries for apt
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
```

Рис. 3: Шаг 2

4 Выводы

Получила навыки правильной работы с репозиториями git.

Список литературы