

Отчёт по лабораторной работе 4

Продвинутое использование git

Бельчуг Александр Константинович

Содержание

1	Цель работы	5
2	Теоретические сведения	6
3	Выполнение лабораторной работы	7
3.1	Работа с тестовым репозиторием	7
4	Вывод	12

Список иллюстраций

3.1	Node.js	7
3.2	установка commitizen	8
3.3	установка standard-changelog	8
3.4	package.json	8
3.5	Отправка	9
3.6	Инициализация и загрузка изменений	9
3.7	Завершение релиза	10
3.8	Отправка	10
3.9	Объединение веток	11
3.10	Завершение релиза	11

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

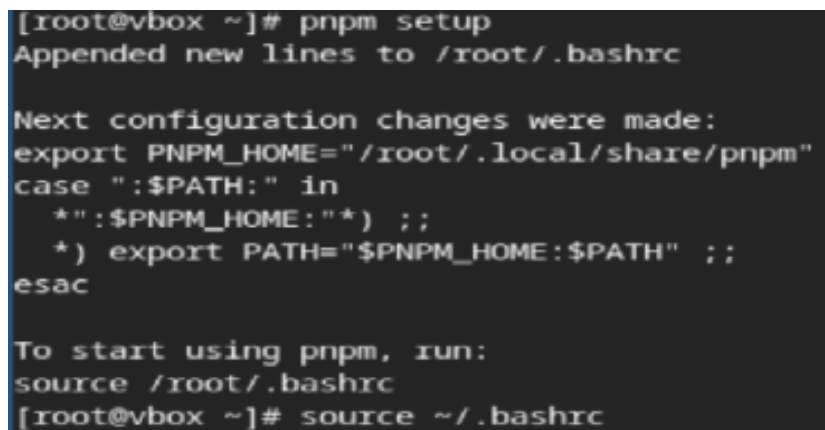
2 Теоретические сведения

- 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

3.1 Работа с тестовым репозиторием

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

A terminal window with a dark background and light-colored text. The text shows the output of the 'pnpm setup' command. It indicates that new lines have been appended to the '/root/.bashrc' file. It then lists the configuration changes made, including setting 'PNPM_HOME' and updating the 'PATH' variable. Finally, it instructs the user to source the '.bashrc' file to start using pnpm.

```
[root@vbox ~]# 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

To start using pnpm, run:
source /root/.bashrc
[root@vbox ~]# source ~/.bashrc
```

Рис. 3.1: Node.js

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

```
[root@vbox ~]# 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
```

Рис. 3.2: установка commitizen

Программа standard-changelog используется для помощи в создании логов.

```
[root@vbox ~]# 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

/root/.local/share/pnpm/global/5:
+ standard-changelog @.0.0
Done in 10.1s
```

Рис. 3.3: установка standard-changelog

Делаем первый коммит и выкладываем на github.

Необходимо заполнить несколько параметров пакета.

Таким образом, файл package.json приобретает вид:

```
/home/akbeljchug/git-test/package.json
{
  "name": "git-test",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "https://github.com:Sasha-860/git-extended.git",
  "author": "Sasha-860 <Sashabelchug1@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.4: package.json

Добавим новые файлы.

Выполним коммит.

Отправим на github.


```
[akbeljchug@vbox git-test]$ git cz
cz:1188 3.1, cz:conventional-changelog@3.2.0

Select the type of change that you're committing: (type: Other changes that don't modify src or test files)
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 72 chars):
() 4
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 21856a9] chore: 4
1 file changed, 14 insertions(+)
create mode 100644 package.json
[akbeljchug@vbox git-test]$ git push
ssh: connect to host github.com port 22: Connection refused
fatal: Не удалось пропустить на внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
к репозиторию существует.
[akbeljchug@vbox git-test]$ git flow init
git: «flow» не является командой git. Смотрите «git --help».

Самые полезные команды:
    setting
    show
```

Рис. 3.5: Отправка

Инициализируем git-flow

Проверьте, что Вы на ветке develop

Загрузите весь репозиторий в хранилище

```
[akbeljchug@vbox git-test]$ 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? []
Hooks and filters directory? [/home/akbeljchug/git-test/.git/hooks]
[akbeljchug@vbox git-test]$
[akbeljchug@vbox git-test]$ git branch
* develop
  main
[akbeljchug@vbox git-test]$ git push --all
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При скатии изменений используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.14 Киб | 1.14 Мб/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Sasha-860/git-extender.git
  63059a1..21856a9  main -> main
  * (new branch)      develop -> develop
[akbeljchug@vbox git-test]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 3.6: Инициализация и загрузка изменений

Установите внешнюю ветку как вышестоящую для этой ветки

Создадим релиз с версией 1.0.0

Создадим журнал изменений

Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

```
[akbeljchug@vbox git-test]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[akbeljchug@vbox git-test]$ 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'

[akbeljchug@vbox git-test]$ standard-changelog --first-release
bash: standard-changelog: команда не найдена
[akbeljchug@vbox git-test]$ standard-changelog --first-release
bash: standard-changelog: команда не найдена
[akbeljchug@vbox git-test]$ standard-changelog --first-release
bash: standard-changelog: команда не найдена
[akbeljchug@vbox git-test]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[akbeljchug@vbox git-test]$ git add CHANGELOG.md
[akbeljchug@vbox git-test]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 1e3b1b0] chore(site): add changelog
1 file changed, 4 insertions(+)
 create mode 100644 CHANGELOG.md
[akbeljchug@vbox git-test]$ git push --all
error: имелось в виду "--all" (с двумя дефисами)?
[akbeljchug@vbox git-test]$ git push --all
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
```

Рис. 3.7: Завершение релиза

Отправим данные на github

```
* [new branch]      release/1.0.0 -> release/1.0.0
[akbeljchug@vbox git-test]$ git push --tags
Everything up-to-date
[akbeljchug@vbox git-test]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Sasha-860/git-extender/releases/tag/v1.2.3
[akbeljchug@vbox git-test]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Sasha-860/git-extender/releases/tag/v1.0.0
[akbeljchug@vbox git-test]$ git flow feature start feature_branch
Переключились на новую ветку «feature/feature_branch»
```

Рис. 3.8: Отправка

Создадим ветку для новой функциональности По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop:

```
[akbeljchug@vbox git-test]$ 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

[akbeljchug@vbox git-test]$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была 21056a9).

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. 3.9: Объединение веток

Создадим релиз с версией 1.2.3

Обновите номер версии в файле package.json. Установите её в 1.2.3

Создадим журнал изменений

Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

```
[akbeljchug@vbox git-test]$ git flow release start 1.2.3
fatal: There is an existing release branch '1.0.0'. Finish that one first.
[akbeljchug@vbox git-test]$ standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[akbeljchug@vbox git-test]$ git add CHANGELOG.md
[akbeljchug@vbox git-test]$ git commit -am 'chore(site): update changelog'
[develop 12e7c6e] chore(site): update changelog
1 file changed, 5 insertions(+)
create mode 100644 CHANGELOG.md
[akbeljchug@vbox git-test]$ git flow release finish 1.2.3
fatal: branch 'release/1.2.3' does not exist and is required.
[akbeljchug@vbox git-test]$ git push --all
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При скачивании изменений используется до 4 потоков
Скачивание объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1017 байтов | 508.00 КиБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Sasha-860/git-extender.git
 21056a9..12e7c6e develop -> develop
[akbeljchug@vbox git-test]$ git push --tags
Everything up-to-date
[akbeljchug@vbox git-test]$ gh release create v1.2.3 -F CHANGELOG.md
HTTP 422: Validation failed (https://api.github.com/repos/Sasha-860/git-extender/releases)
Release tag name already exists
[akbeljchug@vbox git-test]$
```

Рис. 3.10: Завершение релиза

4 Вывод

Мы приобрели практические навыки взаимодействия с дополнительными функциями гитхаб.