

Лабораторная работа №4

Операционные системы

Нелиа Нджову

Содержание

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

Список иллюстраций

3.1	включение репозиторий copr	7
3.2	установки gitflow	7
3.3	установки nodejs	7
3.4	установки pnpm	8
3.5	настройка node.js	8
3.6	перелогинирование	8
3.7	commitizen	8
3.8	standard-changelog	9
3.9	создание новый репозиторий	9
3.10	клонирование новый репозиторий	9
3.11	первый коммитов	10
3.12	создание файла	10
3.13	Конфигурация для пакетов Node.js	11
3.14	добавление файла на git	11
3.15	выполнение коммитов	11
3.16	отправление файла	12
3.17	инициализирование git flow	12
3.18	проверка если я на ветке develop	12
3.19	загрузка весь репозиторий	12
3.20	вышестоящая ветка	12
3.21	создание релиз с версии 1.0.0	13
3.22	создание журнала изменений	13
3.23	дабовление журнала изменнений в индекс	13
3.24	заполнения релизной ветки	14
3.25	отправление на github	14
3.26	создание релиз на github	14
3.27	создание ветки	15
3.28	объединение ветки	15
3.29	создание релиз с версии 1.2.3	15
3.30	обновление номер версии	16
3.31	создание журнала изменений	16
3.32	дабовление журнала изменнений в индекс	16
3.33	заполнение релизной ветки	17
3.34	отправление на github	17
3.35	создание релиз на github	17

Список таблиц

1 Цель работы

Цель работы является 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

1. Установка git-flow

Сначала я включаю репозиторий copr(рис.1),после этого устанавливаю gitflow(рис.2)

```
nelianjovu@nelianjovu:~$ sudo dnf copr enable elegos/gitflow
[sudo] password for nelianjovu:
Enabling a Copr repository. Please note that this repository is not part
of the main distribution, and quality may vary.
```

Рис. 3.1: включение репозиторий copr

```
nelianjovu@nelianjovu:~$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos
Package gitflow-1.12.3-1.fc34.x86_64 is already installed.
```

Рис. 3.2: установки gitflow

2. Установка Node.js

Я устанавливаю nodejs. На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов(рис.3)

```
nelianjovu@nelianjovu:~$ sudo dnf install nodejs
Last metadata expiration check: 0:01:30 ago on Wed 06 Mar 2024 16:48:22 MSK.
```

Рис. 3.3: установки nodejs

Потом я устанавливаю rnrn(рис.4)

```
nelianjovu@nelianjovu:~$ sudo dnf install pnpm
Last metadata expiration check: 0:02:27 ago on Wed 06 Mar 2024 16:48:22 MSK.
```

Рис. 3.4: установки pnpm

3. Настройка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. Я запускаю команду `pnpm setup`. Эта команда автоматически устанавливает yarn в переменную path (рис 5)

```
nelianjovu@nelianjovu:~$ pnpm setup
Appended new lines to /home/nelianjovu/.bashrc

Next configuration changes were made:
```

Рис. 3.5: настройка node.js

Затем я выполняю команду `source ~/.bashrc` (рис.6)

```
nelianjovu@nelianjovu:~$ source ~/.bashrc
```

Рис. 3.6: перелогинирование

4. Общепринятые коммиты

commitizen

Я запускаю команду, которая используется для помощи в форматировании коммитов. При этом устанавливается скрипт `git-cz`, который мы и будем использовать для коммитов (рис.7)

```
nelianjovu@nelianjovu:~$ 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.
```

Рис. 3.7: commitizen

standard-changelog

Я запускаю команду, которая используется для помощи в создании логов(рис.8)

```
nelianjovu@nelianjovu:~$ npm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done
```

Рис. 3.8: standard-changelog

Практический сценарий использования git

Я создаю новый репозиторий под названием git-extended(рис.9)

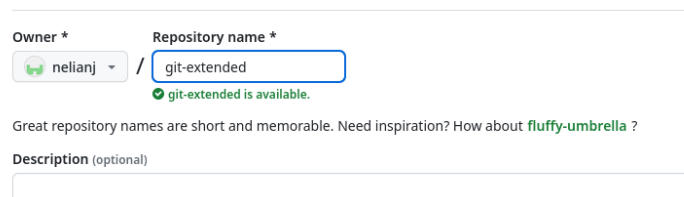


Рис. 3.9: создание новый репозиторий

Я клонирую его в рабочую папку(рис.10)

```
nelianjovu@nelianjovu:~/work$ git clone --recursive https://github.com/nelianj/git-extended.git
Cloning into 'git-extended'...
warning: You appear to have cloned an empty repository.
```

Рис. 3.10: клонирование новый репозиторий

Потом я делаю первый коммитов и выкладываю его на github(рис.11)

```

nelianjovu@nelianjovu:~/work/git-extended$ git add .
nelianjovu@nelianjovu:~/work/git-extended$ git commit -m "first commit"
[main (root-commit) 763fc26] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
nelianjovu@nelianjovu:~/work/git-extended$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 865 bytes | 865.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nelianj/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
nelianjovu@nelianjovu:~/work/git-extended$

```

Рис. 3.11: первый коммитов

Я создаю файл package.json с помощью команды `pnpm init`. Этот файл важен, поскольку он содержит метаданные о проекте(рис.12)

```

nelianjovu@nelianjovu:~/work/git-extended$ pnpm init
Wrote to /home/nelianjovu/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. 3.12: создание файла

Я сконфигурирую формат коммитов. Для этого я добавлю в файл package.json команду для формирования коммитов(рис.13)

```

package.json  [----]  0 L:[ 1+14 15/ 15] *(344 / 344b) <EOF>
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:nelianj/git-extended.git",
  "author": "Njovu Nelia nelianj@gmail.com",
  "license": "CC-BY-4.0"
  "config":{
    "commitizen":{
      "path":"cz-conventional-changelog"
    }
  }
}

```

Рис. 3.13: Конфигурация для пакетов Node.js

Потом добавлю новый файл на github(рис.14)

```

nelianjovu@nelianjovu:~/work/git-extended$ git add .

```

Рис. 3.14: добавление файла на git

Я выполняю коммитов с помощью git cz(рис.15)

```

nelianjovu@nelianjovu:~/work/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):
  (8) add file
? 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 of the commit itself:
-
? Describe the breaking changes:

? Does this change affect any open issues? Yes
? Add issue references (e.g. "fix #123", "re #123").:
123
[main e07a20b] feat: add file
1 file changed, 14 insertions(+)
create mode 100644 package.json

```

Рис. 3.15: выполнение коммитов

Затем я отправляю на github(рис.16)

```
nelianjovu@nelianjovu:~/work/git-extended$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
```

Рис. 3.16: отправление файла

Я инициализирую git flow с помощью команды git flow init(рис.17)

```
nelianjovu@nelianjovu:~/work/git-extended$ git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main] main
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] feature/
Bugfix branches? [bugfix/] bugfix/
Release branches? [release/] release/
Hotfix branches? [hotfix/] hotfix/
Support branches? [support/] support/
Version tag prefix? []
Hooks and filters directory? [/home/nelianjovu/work/git-extended/.git/hooks] /home/nelianjovu/work/git-extended/.git/hoo
```

Рис. 3.17: инициализирование git flow

Я проверяю что я на ветке develop(рис.18)

```
nelianjovu@nelianjovu:~/work/git-extended$ git branch
* develop
main
```

Рис. 3.18: проверка если я на ветке develop

Я загружаю весь репозиторий в хранилище(рис.19)

```
nelianjovu@nelianjovu:~/work/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 by visiting:
remote: https://github.com/nelianjovu/git-extended/pull/new/develop
```

Рис. 3.19: загрузка весь репозиторий

Я устанавливаю внешнюю ветку как вышестоящую для этой ветки(рис.20)

```
nelianjovu@nelianjovu:~/work/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 3.20: вышестоящая ветка

Я создаю релиз с версии 1.0.0(рис.22)

```
nelianjovu@nelianjovu:~/work/git-extended$ git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
```

Рис. 3.21: создание релиз с версии 1.0.0

Потом я создаю журнал изменений(рис.23)

```
nelianjovu@nelianjovu:~/work/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 3.22: создание журнала изменений

После этого я дабовлю журнал изменений в индекс(рис.24)

```
nelianjovu@nelianjovu:~/work/git-extended$ git add CHANGELOG.md
nelianjovu@nelianjovu:~/work/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 db8ac5c] chore(site): add changelog
1 file changed, 10 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.23: дабование журнала изменнений в индекс

Я заполняю релизную ветку в основную ветку(рис.25)

```

nelianjovu@nelianjovu:~/work/git-extended$ git flow release finish 1.0.0
Branches 'main' and 'origin/main' have diverged.
And local branch 'main' is ahead of 'origin/main'.
Already on 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
 CHANGELOG.md | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was db8ac5c).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged '1.0.0'
- Release tag '1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'

```

Рис. 3.24: заполнения релизной ветки

Я отправляю данные на github(рис.26)

```

nelianjovu@nelianjovu:~/work/git-extended$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.82 KiB | 192.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nelianj/git-extended.git
 c0a7bdc..bea736d develop -> develop
 c0a7bdc..d397cc6 main -> main
nelianjovu@nelianjovu:~/work/git-extended$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 154 bytes | 154.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nelianj/git-extended.git
 * [new tag]         1.0.0 -> 1.0.0

```

Рис. 3.25: отправление на github

Я создаю релиз на github.Для этого буду использовать утилиты работы с github
“gh release create v1.0.0 -F CHANGELOG.md”(рис.27)

```

nelianjovu@nelianjovu:~/work/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/nelianj/git-extended/releases/tag/v1.0.0

```

Рис. 3.26: создание релиз на github

Я создаю ветку для новой функциональности(рис.28)

```
nelianjovu@nelianjovu:~/work/git-extended$ git flow feature start feature_branch
Switched to a new branch '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'
```

Рис. 3.27: создание ветки

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop(рис.28)

```
nelianjovu@nelianjovu:~/work/git-extended$ git flow feature finish feature_branch
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Already up to date.
Deleted branch feature/feature_branch (was e6af570).
```

Рис. 3.28: объединение ветки

Я создаю релиз с версией 1.2.3(рис.29)

```
nelianjovu@nelianjovu:~/work/git-extended$ git flow release start 1.2.3
Switched to a new branch '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!
```

Рис. 3.29: создание релиз с версией 1.2.3

Я обновлю номер версии в файле package.json. Устанавливаю её в 1.2.3(рис.30)

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for
  "main": "index.js",
  "repository": "git@github.com:
  "author": "Njovu Nelia nelian
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
```

Рис. 3.30: обновление номер версии

Я создаю журнал изменений(рис.31)

```
nelianjovu@nelianjovu:~/work/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
nelianjovu@nelianjovu:~/work/git-extended$
```

Рис. 3.31: создание журнала изменений

Я добавляю журнал изменений в индекс(рис.32)

```
nelianjovu@nelianjovu:~/work/git-extended$ git add CHANGELOG.md
nelianjovu@nelianjovu:~/work/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 2d35f2c] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
nelianjovu@nelianjovu:~/work/git-extended$
```

Рис. 3.32: добавление журнала изменений в индекс

Заполняю релизную ветку в основную ветку(рис.33)


```
nelianjovu@nelianjovu:~/work/git-extended$ git flow release finish 1.2.3
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Already on 'main'
```

Рис. 3.33: заполнение релизной ветки

Я отправляю данные на github(рис.34)

```
nelianjovu@nelianjovu:~/work/git-extended$ git push --all
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.77 KiB | 354.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/nelianj/git-extended.git
    bea736d..2f2c2b7 develop -> develop
    d397cc6..3925bcd main -> main
nelianjovu@nelianjovu:~/work/git-extended$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 158 bytes | 158.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nelianj/git-extended.git
    * [new tag]         1.2.3 -> 1.2.3
```

Рис. 3.34: отправление на github

Я создаю релиз на github с комментарием из журнала изменений(рис.35)

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

Рис. 3.35: создание релиз на github

4 Выводы

Выполняя эту лабораторную работу, я получаю навыки правильной работы с репозиториями git.

Список литературы

Лабораторная работа № 4