

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров и операционные системы

Мутале Чали

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы	1
3.1	Установка git-flow	1
3.2	Установка Node.js	2
3.3	Настройка Node.js	2
3.4	Общепринятые коммиты	2
4	Выводы	8
	Список литературы	8

1 Цель работы

Цель данной работы является получением 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 git-flow

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Сначала я включаю репозиторий с помощью команды:

```
cmutale@kali:~$ sudo dnf copr enable elegos/gitflow
[sudo] password for cmutale:
https://copr.fedorainfracloud.org/api_3/rpmrepo/elego 100% | 351.0 B/s | 652.0 B | 00m02s
Enabling a Copr repository. Please note that this repository is not part
of the main distribution, and quality may vary.
```

Рис. 1: включение copr

Используя dnf install скачаю gitflow:

```
cmutale@kali:~$ sudo dnf install gitflow
Updating and loading repositories:
RPM Fusion for Fedora 41 - Nonfree - NVIDIA Driver 100% | 5.7 KiB/s | 8.8 KiB | 00m02s
RPM Fusion for Fedora 41 - Nonfree - Steam 100% | 23.5 KiB/s | 8.7 KiB | 00m00s
```

Рис. 2: dnf install gitflow

3.2 Установка Node.js

Для семантического версионирования и общепринятых коммитов я устанавливаю Nodejs и pnpm:

```
cmutale@kali:~$ sudo dnf install pnpm
Updating and loading repositories:
```

Рис. 3: Установка Nodejs

```
cmutale@kali:~$ pnpm setup
```

Рис. 4: Установка pnpm

3.3 Настройка Node.js

Запуская pnpm setup я добавляю каталог с исполняемыми файлами, устанавливаемыми уагн для работы с Node.js в переменную PATH:

```
To start using pnpm, run:
source /home/cmutale/.bashrc
cmutale@kali:~$ ^C
cmutale@kali:~$ source /home/cmutale/.bashrc
```

Рис. 5: Запуск pnpm

Далее перелогинюсь и выполняю source ~/.bashrc:

```
To start using pnpm, run:
source /home/cmutale/.bashrc
cmutale@kali:~$ ^C
cmutale@kali:~$ source /home/cmutale/.bashrc
```

Рис. 6: ~/.bashrc

3.4 Общепринятые коммиты

Для помощи в форматировании коммитов добавляю программу commitizen:

```
cmutale@kali:~$ pnpm add -g commitizen
Progress: resolved 13, reused 0, downloaded 8, added 0
```

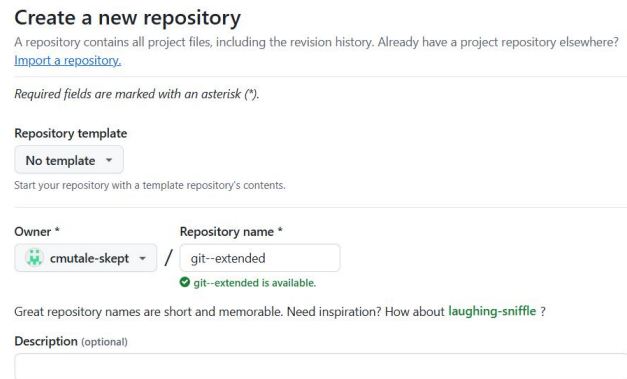
Рис. 7: добавление commitizen

Добавляю standard-changelog для помощи в создании логов:

```
cmutale@kali:~$ npm add -g standard-changelog
Progress: resolved 19, reused 14, downloaded 0, added 0
```

Рис. 8: добавление standard-changelog

Создаю репозиторий на GitHub назову его git-extended:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner *

Repository name *

cmutale-skept / git-extended

git-extended is available.

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

Description (optional)

Рис. 9: Создание git-extended

Я клонирую его в каталог git-extended:

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

Рис. 10: клонирование git-extended

Создаю и добавляю файл README.md:

```
cmutale@kali:~$ cd git-extended
cmutale@kali:~/git-extended$ git add README.md
fatal: pathspec 'README.md' did not match any files
cmutale@kali:~/git-extended$ touch README.md
cmutale@kali:~/git-extended$ git add README.md
cmutale@kali:~/git-extended$
```

Рис. 11: Добавление readme.md

Делаю первый коммит и выкладываю на github:

```
cmutale@kali:~/git-extended$ git commit -m "first message"
[main (root-commit) 7c80668] first message
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
cmutale@kali:~/git-extended$ git remote add origin github.com/cmutale-skept/git-extended.git
error: remote origin already exists.
cmutale@kali:~/git-extended$ git remote -v
origin https://github.com/cmutale-skept/git-extended.git (fetch)
origin https://github.com/cmutale-skept/git-extended.git (push)
cmutale@kali:~/git-extended$ git push -u origin master
```

Рис. 12: Первый коммит

Я инициализирую npm:

```

cmutale@kali:~/git-extended$ npm init
Wrote to /home/cmutale/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. 13: Инициализирование npm

Заполняю несколько параметров пакета (файл package.json):

```

GNU nano 8.1 cmutale@kali:~/git-extended — nano package.json Modified
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:cmutale-skept/git-extended.git",
  "author": "cmutale <1032239667@pfur.ru>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 14: Заполнение пакетов

Добавляю новые файлы и выполняю коммит (указываю тип коммит (feat)) и отправляю на гит:

```

cmutale@kali:~/git-extended$ git add .
cmutale@kali:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: (Use arrow keys)
> feat: A new feature
  fix: A bug fix
  docs: Documentation only changes
  style: Changes that do not affect the meaning of the code (white-space, formatting,
missing semi-colons, etc)
  refactor: A code change that neither fixes a bug nor adds a feature
  perf: A code change that improves performance
(Move up and down to reveal more choices)

```

Рис. 15: Добавление новых файлов

Инициализирую git-flow и указываю ветки:

```

cmutale@kali:~/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/cmutale/git-extended/.git/hooks]

```

Рис. 16: Инициализирование git-flow

Проверяю что я на ветке develop с помощью git branch:

```

cmutale@kali:~/git-extended$ git branch
* develop
main

```

Рис. 17: Проверка ветки

Загружаю весь репозиторий в хранилище с помощью git push --all:

```

cmutale@kali:~/git-extended$ git push --all
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 3 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.15 KiB | 393.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/cmutale-skept/git-extended.git
  7c80668..4098290  main -> main
  * [new branch]      develop -> develop

```

Рис. 18: Загрузка репозиторий

Устанавливаю внешнюю ветку как вышестоящую для этой ветки (develop):

```

cmutale@kali:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 19: установка внешней ветки

Создаю релиз с версией 1.0.0 и журнал изменений (standard-changelog)::

```

cmutale@kali:~/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'
- 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'

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

```

Рис. 20: Создание релиза

Добавляю журнал изменений в индекс:

```
cmutale@kali:~/git-extended$ git add CHANGELOG.md
cmutale@kali:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 89084d9] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 21: Добавление журнала изменений

Залью релизную ветку в основную ветку:

```
cmutale@kali:~/git-extended$ git flow release finish 1.0.0
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
To https://github.com/cmudale-skept/git-extended.git
- [deleted]          release/1.0.0
Deleted branch release/1.0.0 (was 89084d9).

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; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
```

Рис. 22: Замена ветки

Отправляю данные на github:

```
cmutale@kali:~/git-extended$ git push --all
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 941 bytes | 313.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/cmudale-skept/git-extended.git
4098290..8bf9ab4 develop -> develop
* [new branch]      feature/feature_branch -> feature/feature_branch
cmutale@kali:~/git-extended$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 193 bytes | 193.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/cmudale-skept/git-extended.git
* [new tag]         1.0.0 -> 1.0.0
cmutale@kali:~/git-extended$
```

Рис. 23: Отправка на git

Создаю релиз на github. Для этого использую утилиты работы с github gh (gh release create):

```
cmutale@kali:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/cmudale-skept/git-extended/releases/tag/v1.0.0
cmutale@kali:~/git-extended$
```

Рис. 24: Создание релиза

Создаю ветку для новой функциональности:

```

cmutale@kali:~/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'

Now, start committing on your feature. When done, use:

git flow feature finish feature_branch

```

Рис. 25: Создание ветки *feature_branch*

Далее, продолжаю работу с git как обычно. Создаю релиз с версией 1.2.3:

```

cmutale@kali:~/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!
- Start committing last-minute fixes in preparing your release
- When done, run:

git flow release finish '1.2.3'

```

Рис. 26: Создание релиза с версией 1.2.3

Обновляю номер версии в файле package.json в 1.2.3:

```

package.json  [----] 0 L: [ 1+ 0 1/ 17] *(0 / 353b) 012
{
  "name": "git-extended",
  "version": "1.2.3",

```

Рис. 27: Обновление номера версии

Создаю журнал изменений (standard-changelog):

```

cmutale@kali:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
cmutale@kali:~/git-extended$

```

Рис. 28: Создание нового журнала изменений

Добавляю журнал изменений в индекс:

```

cmutale@kali:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
cmutale@kali:~/git-extended$ git add CHANGELOG.md
cmutale@kali:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 3a0d086] chore(site): update changelog
1 file changed, 4 insertions(+)

```

Рис. 29: Добавление журнала

Залью релизную ветку в основную ветку:

```

cmutale@kali:~/git-extended$ git flow release finish 1.2.3
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 3 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 | 4 ++++
1 file changed, 4 insertions(+)
Deleted branch release/1.2.3 (was 3a0d086).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged '1.2.3'
- Release tag '1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

```

Рис. 30: Замена ветки

Отправляю данные на github:

```

cmutale@kali:~/git-extended$ git push --all
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 3 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.74 KiB | 1.37 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/cmutale-skept/git-extended.git
8bf9ab4..2e62d62 develop -> develop
c59b3fa..8bb8b4a main -> main

```

Рис. 31: Отправка данных

Создаю релиз на github с комментарием из журнала изменений:

```

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

```

Рис. 32: Создание релиза

4 Выводы

При выполнении работы я получила навыки правильной работы с репозиториями git.

Список литературы

Архитектура ЭВМ