

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров и операционные системы

Джаллох Ишмаил

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка git-flow	6
3.2	Установка Node.js	6
3.3	Настройка Node.js	7
3.4	Общепринятые коммиты	7
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	включение copr	6
3.2	dnf install gitflow	6
3.3	Установка Nodejs	6
3.4	Установка pnpm	6
3.5	Запуск pnpm	7
3.6	~/bashrc	7
3.7	добавление commitizen	7
3.8	добавление standard-changelog	7
3.9	Создание git-extended	8
3.10	клонирование git-extended	8
3.11	Добавление readme.md	8
3.12	Первый коммит	8
3.13	Инициализирование pnpm	9
3.14	Заполнение пакетов	9
3.15	Добавление новых файлов	9
3.16	Инициализирование git-flow	10
3.17	Проверка ветки	10
3.18	Загрузка репозиторий	10
3.19	установка внешней ветки	10
3.20	Создание релиза	11
3.21	Добавление журнала изменений	11
3.22	Замена ветки	11
3.23	Отправка на гит	12
3.24	Создание релиза	12
3.25	Создание ветки feature_branch	12
3.26	Создание релиза с версией 1.2.3	12
3.27	Обновление номера версии	13
3.28	Создание нового журнала изменений	13
3.29	Добавление журнала	13
3.30	Замена ветки	13
3.31	Отправка данных	14
3.32	Создание релиза	14

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 copr:

```
ijalloheijalloh:~$ sudo dnf copr enable elegos/gitflow
[sudo] password for ijalloh:
https://copr.fedorainfracloud.org/api 100% | 206.0 B/s | 652.0 B | 00m03s
Enabling a Copr repository. Please note that this repository is not part
```

Рис. 3.1: включение copr

Используя dnf install скачаю gitflow:

```
ijalloheijalloh:~$ sudo dnf install gitflow
Updating and loading repositories:
Fedora 41 - x86_64 - Updates      100% | 7.3 KiB/s | 25.9 KiB | 00m04s
Fedora 41 - x86_64 - Updates      100% | 112.9 KiB/s | 2.6 MiB | 00m24s
```

Рис. 3.2: dnf install gitflow

3.2 Установка Node.js

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

```
ijalloheijalloh:~$ sudo dnf install nodejs
Updating and loading repositories:
Repositories loaded.
```

Рис. 3.3: Установка Nodejs

```
ijalloheijalloh:~$ sudo dnf install npm
Updating and loading repositories:
Repositories loaded.
```

Рис. 3.4: Установка npm

3.3 Настройка Node.js

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

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

Рис. 3.5: Запуск `pnpm`

Далее перелогинюсь и выполняю `source ~/.bashrc`:

```
To start using pnpm, run:
source /home/ijalloh/.bashrc
ijalloh@ijalloh:~$ source /home/ijalloh/.bashrc
```

Рис. 3.6: `~/.bashrc`

3.4 Общепринятые коммиты

Для помощи в форматировании коммитов добавляю программу `commitizen`:

```
ijalloh@ijalloh:~$ pnpm add -g commitizen
```

Рис. 3.7: добавление `commitizen`

Добавляю `standard-changelog` для помощи в создании логов:

```
ijalloh@ijalloh:~$ 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
```

Рис. 3.8: 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 *

Isho-Jah

Repository name *

git-extended

git-extended is available.

Great repository names are short and memorable. Need inspiration? How about [bug-free-waddle](#) ?

Рис. 3.9: Создание git-extended

Я клонирую его в каталог git-extended:

```
ijalloheijalloh:~$ git clone --recursive https://github.com/Isho-Jah/git-extended.git git-extended
fatal: destination path 'git-extended' already exists and is not an empty directory
```

Рис. 3.10: клонирование git-extended

Создаю и добавляю файл README.md:

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

Рис. 3.11: Добавление readme.md

Делаю первый коммит и выкладываю на github:

```
ijalloheijalloh:~/git-extended$ git commit -m "first commit"
[main (root-commit) ff6bc7a] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
ijalloheijalloh:~/git-extended$ git remote add origin git@github.com:Isho-Jah/git-extended.git
error: remote origin already exists.
ijalloheijalloh:~/git-extended$ git remote -v
origin https://github.com/Isho-Jah/git-extended.git (fetch)
origin https://github.com/Isho-Jah/git-extended.git (push)
ijalloheijalloh:~/git-extended$ git push -u origin master
```

Рис. 3.12: Первый коммит

Я инициализирую рпм:


```

ijalloh@ijalloh:~/git-extended$ pnpm init
Wrote to /home/ijalloh/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"
}
ijalloh@ijalloh:~/git-extended$

```

Рис. 3.13: Инициализирование pnpm

Заполняю несколько параметров пакета (файл package.json):

```

package.json  [-M--] 3 L:[ 1+12 13/ 15] *(349 / 356b) 0032 0x020 [*][X]
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Isho-Jah/git-extended.git",
  "author": "Isho-Jah <isjalloh6040@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 3.14: Заполнение пакетов

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

```

ijalloh@ijalloh:~/git-extended$ git add .
ijalloh@ijalloh:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0
? Select the type of change that you're committing: feat:  A new feature

```

Рис. 3.15: Добавление новых файлов

Инициализирую git-flow и указываю ветки:

```

ijalloh@ijalloh:~/git-extended$ git flow init
Already initialized for gitflow.
To force reinitialization, use: git flow init -f
ijalloh@ijalloh:~/git-extended$ git flow init -f

Which branch should be used for bringing forth production releases?
- develop
- master
Branch name for production releases: [master] master

Which branch should be used for integration of the "next release"?
- develop
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/ijalloh/git-extended/.git/hooks]

```

Рис. 3.16: Инициализирование git-flow

Проверяю что я на ветке develop с помощью git branch:

```

ijalloh@ijalloh:~/git-extended$ git branch
* develop
  master

```

Рис. 3.17: Проверка ветки

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

```

ijalloh@ijalloh:~/git-extended$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Isho-Jah/git-extended/pull/new/develop
remote:
To github.com:Isho-Jah/git-extended.git
* [new branch]      develop -> develop

```

Рис. 3.18: Загрузка репозиторий

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

```

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

```

Рис. 3.19: установка внешней ветки

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

```

ijalloh@ijalloh:~/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'

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

```

Рис. 3.20: Создание релиза

Добавляю журнал изменений в индекс:

```

ijalloh@ijalloh:~/git-extended$ git add CHANGELOG.md
ijalloh@ijalloh:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 454a830] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md

```

Рис. 3.21: Добавление журнала изменений

Залью релизную ветку в основную ветку:

```

ijalloh@ijalloh:~/git-extended$ git flow release finish 1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Already on 'master'
Your branch is ahead of 'origin/master' 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 | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was 454a830).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- 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. 3.22: Замена ветки

Отправляю данные на github:

```

ijalloh@ijalloh:~/git-extended$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.80 KiB | 955.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Isho-Jah/git-extended.git
   bd4b24b..8efc236  develop -> develop
   bd4b24b..be0a8e7  master  -> master

```

Рис. 3.23: Отправка на гит

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

```

ijalloh@ijalloh:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Isho-Jah/git-extended/releases/tag/v1.0.0

```

Рис. 3.24: Создание релиза

Создаю ветку для новой функциональности:

```

ijalloh@ijalloh:~/git-extended$ git flow release start feature_branch
Switched to a new branch 'release/feature_branch'

Summary of actions:
- A new branch 'release/feature_branch' was created, based on 'develop'
- You are now on branch 'release/feature_branch'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish 'feature_branch'

```

Рис. 3.25: Создание ветки feature_branch

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

```

ijalloh@ijalloh:~/git-extended$ git flow release start 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
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. 3.26: Создание релиза с версией 1.2.3

Обновляю номер версии в файле package.json в 1.2.3:

```
package.json  [-M--] 20 L:[ 1+ 2 3/ 15] *(49 / 369b) 0034 0x022 [
{
  "name": "git-extended",
  "version": "1.2.3",
}
```

Рис. 3.27: Обновление номера версии

Создаю журнал изменений (standard-changelog):

```
^[[Aijalloh@ijalloh:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
```

Рис. 3.28: Создание нового журнала изменений

Добавляю журнал изменений в индекс:

```
ijalloh@ijalloh:~/git-extended$ git add CHANGELOG.md
ijalloh@ijalloh:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 3656596] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 3.29: Добавление журнала

Залью релизную ветку в основную ветку:

```
ijalloh@ijalloh:~/git-extended$ git flow release finish 1.2.3
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Already on 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is ahead of 'origin/develop' by 2 commits.
(use "git push" to publish your local commits)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Deleted branch release/1.2.3 (was 3656596).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'master'
- 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. 3.30: Замена ветки

Отправляю данные на github:

```

ijalloh@ijalloh:~/git-extended$ git push --all
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 4.50 KiB | 4.50 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Isho-Jah/git-extended.git
   8efc236..76ad4b9  develop -> develop
   be0a8e7..6e5ef2c  master -> master
ijalloh@ijalloh:~/git-extended$ git push --tags

```

Рис. 3.31: Отправка данных

Создаю релиз на github с комментарием из журнала изменений:

```

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

```

Рис. 3.32: Создание релиза

4 Выводы

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

Список литературы

Архитектура ЭВМ