

Отчет по лабораторной работе №4

Дисциплина: Операционные системы

Кириянова Екатерина Андреевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Git-flow	7
4.2	Git-flow	7
4.3	Node.js	7
4.4	Node.js	7
4.5	Запуск	8
4.6	Выполнение	8
4.7	Программа	8
4.8	Скрипт	9
4.9	Коммит	9
4.10	Конфигурация	9
4.11	Файл	10
4.12	Действия	10
4.13	GitHub	10
4.14	Инициализация	11
4.15	Ветка	11
4.16	Загрузка	11
4.17	Создание	11
4.18	Создание	11
4.19	Журнал	12
4.20	GitHub	12
4.21	Релиз	12
4.22	Ветка	12
4.23	Объединение	12
4.24	Релиз	12
4.25	GitHub	13

1 Цель работы

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 Теоретическое введение

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.

4 Выполнение лабораторной работы

Ввожу команду `dnf copr enable elegos/gitflow` (рис. 4.1).

```
eakiryanova@eakiryanova:~$ sudo dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_ 100% | 971.0  B/s | 652.0  B | 00m01s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является
частью
основного дистрибутива, и его качество может отличаться.
```

Рис. 4.1: Git-flow

Ввожу команду `dnf install gitflow` (рис. 4.2).

```
eakiryanova@eakiryanova:~$ sudo dnf install gitflow
Обновление и загрузка репозитория:
Copr repo for gitflow owned by elegos 100% | 2.3 KiB/s | 2.4 KiB | 00m01s
```

Рис. 4.2: Git-flow

Ввожу команду `dnf install nodejs` (рис. 4.3).

```
eakiryanova@eakiryanova:~$ sudo dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
```

Рис. 4.3: Node.js

Ввожу команду `dnf install pnpm` (рис. 4.4).

```
eakiryanova@eakiryanova:~$ sudo dnf install pnpm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
pnpm noarch 9.13.0-1.fc41 updates 15.5 MiB
```

Рис. 4.4: Node.js

Запускаю pnpm setup (рис. 4.5).

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

Next configuration changes were made:
export PNPM_HOME="/home/eakiryanova/.local/share/pnpm"
case ":$PATH:" in
  *:$PNPM_HOME:*) ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/eakiryanova/.bashrc
```

Рис. 4.5: Запуск

Выполняю source ~/.bashrc (рис. 4.6).

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

Рис. 4.6: Выполнение

Команда для программы, которая используется для помощи в форматировании коммитов (рис. 4.7).

```
eakiryanova@eakiryanova:~$ 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

/home/eakiryanova/.local/share/pnpm/global/5:
+ commitizen 4.3.1
```

Рис. 4.7: Программа

Устанавливаю скрипт (рис. 4.8).


```
eakiryanova@eakiryanova:~$ 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 34, added 30
```

Рис. 4.8: Скрипт

Делаю первый коммит и выкладываю на GitHub (рис. 4.9).

```
eakiryanova@eakiryanova:~/git-extended$ git remote add origin https://github.com
/scxkl/git-extended.git
eakiryanova@eakiryanova:~/git-extended$ git branch -M main
eakiryanova@eakiryanova:~/git-extended$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 930 байтов | 232.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/scxkl/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 4.9: Коммит

Провожу конфигурацию для пакетов Node.js (рис. 4.10).

```
eakiryanova@eakiryanova:~/git-extended$ pnpm init
Wrote to /home/eakiryanova/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. 4.10: Конфигурация

Добавляю в файл команду для коммитов (рис. 4.11).

```
package.json [----] 0 L: [ 1+ 0 1/ 15] *(0 / 385b)
{
  "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"
    }
  }
}
```

Рис. 4.11: Файл

Добавляю новые файлы и выполняю коммит (рис. 4.12).

```
eakiryanova@eakiryanova:~/git-extended$ git add .
eakiryanova@eakiryanova:~/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
? What is the scope of this change (e.g. component or file name): (press enter to skip)
```

Рис. 4.12: Действия

Отправляю на GitHub (рис. 4.13).

```
eakiryanova@eakiryanova:~/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.18 КиБ | 201.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/scxkl/git-extended.git
42743d7..3df6fa9 main -> main
```

Рис. 4.13: GitHub

Инициализирую git-flow (рис. 4.14).

```
eakiryanova@eakiryanova:~/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/
```

Рис. 4.14: Инициализация

Проверяю ветку (рис. 4.15).

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

Рис. 4.15: Ветка

Загружаю весь репозиторий в хранилище (рис. 4.16).

```
eakiryanova@eakiryanova:~/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/scxkl/git-extended/pull/new/develop
remote:
To https://github.com/scxkl/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 4.16: Загрузка

Создаю релиз с версией 1.0.0 (рис. 4.17).

```
eakiryanova@eakiryanova:~/git-extended$ git flow release start 1.0.0
```

Рис. 4.17: Создание

Создаю журнал изменений (рис. 4.10).

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

Рис. 4.18: Создание

Добавляю журнал изменений в индекс (рис. 4.19).

```
eakiryanova@eakiryanova:~/git-extended$ git add CHANGELOG.md
```

Рис. 4.19: Журнал

Отправляю данные на GitHub (рис. 4.20).

```
eakiryanova@eakiryanova:~/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 2.03 КиБ | 2.03 МиБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/scxkl/git-extended.git
 3df6fa9..4ccc9f5  main -> main
 * [new branch]      release/1.0.0 -> release/1.0.0
eakiryanova@eakiryanova:~/git-extended$ git push --tags
Everything up-to-date
```

Рис. 4.20: GitHub

Создаю релиз на github (рис. 4.21).

```
eakiryanova@eakiryanova:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/scxkl/git-extended/releases/tag/v1.0.0
```

Рис. 4.21: Релиз

Создаю ветку для новой функциональности (рис. 4.22).

```
eakiryanova@eakiryanova:~/git-extended$ git flow feature start feature_branch
```

Рис. 4.22: Ветка

Объединяю ветку feature_branch с develop (рис. 4.23).

```
eakiryanova@eakiryanova:~/git-extended$ git flow feature finish feature_branch
```

Рис. 4.23: Объединение

Создаю релиз с версией 1.2.3 (рис. 4.24).

```
eakiryanova@eakiryanova:~/git-extended$ git flow release start 1.2.3
```

Рис. 4.24: Релиз

Заливаю релизную ветку в основную и отправляю данные на GitHub (рис. 4.25).

```
eakiryanova@eakiryanova:~/git-extended$ git push --all
Everything up-to-date
eakiryanova@eakiryanova:~/git-extended$ git push --tags
Everything up-to-date
```

Рис. 4.25: GitHub

5 Выводы

В ходе выполнения данной лабораторной работы я получила навыки правильной работы с репозиториями git.

Список литературы

1.Операционные системы