

Отчёт по лабораторной работе №4

Операционные системы

Федорова Анжелика Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16
6	Список литературы	17

Список иллюстраций

Список таблиц

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 Дриссеном. 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

Выполняем установку из коллекции репозитория Corp (рис.1 и рис.2).

```
root@aifedorova:~# dnf copr enable elegos/gitflow
root@aifedorova:~# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:20 назад, Чт 29 фев
2024 00:05:42.
Зависимости разрешены.
```

Устанавливаю пакеты Node.js, на которых базируется программное обеспечение для семантического версионирования и общепринятых коммитов (рис.3).

```
root@aifedorova:~# apt-get install pnpm
```

Запускаю нужную команду pnpm setup (рис.4)

```
root@aifedorova:~# 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
```

Выполняю команду source ~/.bashrc, для того, чтобы работать с окружением для git-flow. (рис.5)

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

Запускаю команду commitizen для помощи в форматировании коммитов.(рис.6)


```

aifedorova@aifedorova:~$ pnpm add -g commitizen
Packages: +152
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/home/aifedorova/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 40.8s

```

Выполняю команду standard-changelog для помощи в создании логов.(рис.7)

```

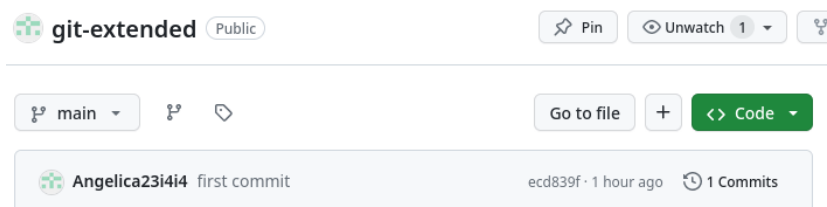
aifedorova@aifedorova:~$ pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/home/aifedorova/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0

Done in 43.4s
aifedorova@aifedorova:~$

```

Теперь создаю репозиторий git-extended на Github(рис.8)



Теперь создаю репозиторий git-extended на консоли и захожу туда (рис.9)

```

aifedorova@aifedorova:~/work$ mkdir git-extended
aifedorova@aifedorova:~/work$ cd git-extended/

```

Создаю первый файл README.md и инициализирую репозиторий(рис.10)

```

aifedorova@aifedorova:~/work/git-extended$ echo "# git-extended" >> README.md
aifedorova@aifedorova:~/work/git-extended$ ls
README.md
aifedorova@aifedorova:~/work/git-extended$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:     git config --global init.defaultBranch <name>
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/aifedorova/work/git-extended/.git/
aifedorova@aifedorova:~/work/git-extended$ git add README.md

```

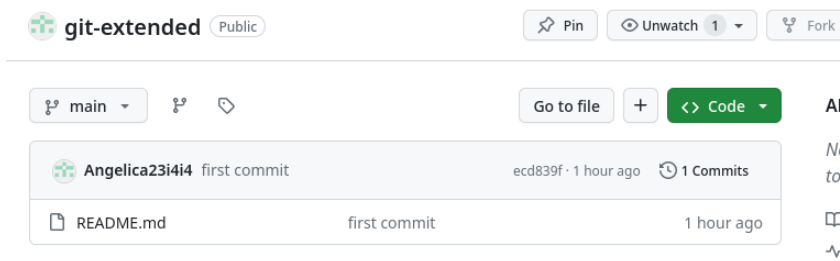
Совершаю первый коммит с помощью команды `git commit -m "first commit"` и подключаю удаленный репозиторий к консоли с помощью команды `git remote add origin` (ссылка на репозиторий). Затем отправляю изменения в репозиторий на сайте. (рис.11)

```

aifedorova@aifedorova:~/work/git-extended$ git commit -m "first commit"
[master (корневой коммит) ecd839f] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
aifedorova@aifedorova:~/work/git-extended$ git branch -M main
aifedorova@aifedorova:~/work/git-extended$ git remote add origin https://github.com/Angelica23i4i4/git-extended.git
aifedorova@aifedorova:~/work/git-extended$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 226 байтов | 113.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Angelica23i4i4/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

Вижу, что изменения были успешно отправлены (рис.12)



Создаю файл package.json с помощью команды `pnpm init` (рис.13)

```

aifedorova@aifedorova:~/work/git-extended$ pnpm init
Wrote to /home/aifedorova/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"
}
aifedorova@aifedorova:~/work/git-extended$ ls
package.json  README.md
aifedorova@aifedorova:~/work/git-extended$

```

Изменяю лицензию в файле на CC-BY-4.0 и добавляю код с `config` как было дано в лабораторной работе(рис.14)

```

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "repository": "https://github.com/Angelica23i4i4/git-extended.git",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "aifedorova iigggooo666@gmail.com",
  "license": "CC-BY-4.0"
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
aifedorova@aifedorova:~/work/git-extended$

```

Добавлю новые файлы с помощью `git add` и выполню коммит с помощью `git cz`(рис.15)

```

aifedorova@aifedorova:~/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: style:   Changes that do not affect the meaning of the code
(white-space, formatting, missing semi-colons, etc)
? 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 93 chars):
(4) code
? 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 fc95ef9] style: code

```

Отправляю изменения в локальный репозиторий(рис.16)

```

aifedorova@aifedorova:~/work/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 492 байта | 492.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Angelica23i4i4/git-extended.git
   ecd839f..fc95ef9  main -> main
aifedorova@aifedorova:~/work/git-extended$

```

Для работы с конфигурацией git-flow я сначала инициализирую git-flow с git flow init. Префикс для ярлыков установим в v(рис.17)

```

aifedorova@aifedorova:~/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/] v
Bugfix branches? [bugfix/] c
Release branches? [release/] v
Hotfix branches? [hotfix/] v
Support branches? [support/] v
Version tag prefix? [] v
Hooks and filters directory? [/home/aifedorova/work/git-extended/.git/hooks] /home/aifedorova/work/git-extended/.git/hooks

```

Проверяю , что я нахожусь на ветке develop (рис.18)

```

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

```

Загружаю весь репозиторий в хранилище с помощью git push --all (рис.19)

```

aifedorova@aifedorova:~/work/git-extended$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Angelica23i4i4/git-extended/pull/new/develop
remote:
To https://github.com/Angelica23i4i4/git-extended.git
 * [new branch]      develop -> develop

```

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

```

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

```

Создаю релиз с версией 1.0.0 (рис.21)

```

aifedorova@aifedorova:~/work/git-extended$ git flow release start 1.0.0
Переключились на новую ветку «v1.0.0»

Summary of actions:
- A new branch 'v1.0.0' was created, based on 'develop'
- You are now on branch 'v1.0.0'

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'

```

Создам журнал изменений (рис.22)

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

Добавлю журнал изменений в индекс (рис.23)

```
aifedorova@aifedorova:~/work/git-extended$ git add CHANGELOG.md
aifedorova@aifedorova:~/work/git-extended$ git commit -am 'chore(site): add changelog'
[v1.0.0 5064ca9] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
```

Теперь я должна положить релизную ветку в основную ветку. Далее я ввожу необходимые сообщения для обозначения цели изменений (рис.24)

```
Merge branch 'v1.0.0'
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной вышестоящей
# ветки в тематическую ветку.
#
# Строки, начинающиеся с «#» будут проигнорированы, а пустое сообщение
# отменяет процесс коммита.
```

Вижу, что данный процесс был успешно завершен (рис.25)

```
aifedorova@aifedorova:~/work/git-extended$ git flow release finish 1.0.0
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
warning: refname 'v1.0.0' is ambiguous.
warning: refname 'v1.0.0' is ambiguous.
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Ветка v1.0.0 удалена (была 5064ca9).
```

Далее я отправляю данные на github (рис.26)

```
aifedorova@aifedorova:~/work/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 547 байтов | 547.00 КиБ/с, готово.
Всего 5 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Angelica23i4i4/git-extended.git
fc95ef9..761002d develop -> develop
fc95ef9..b9ceb72 main -> main
```

Затем отправляю в репозиторий все предоставленные теги (рис.27)

```
aifedorova@aifedorova:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байтов | 155.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Angelica23i4i4/git-extended.git
* [new tag] v1.0.0 -> v1.0.0
```

Создам релиз на github. Для этого буду использовать утилиты работы с github (рис.28)

```
aiifedorova@aifedorova:~/work/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Angelica23i4i4/git-extended/releases/tag/v1.0.0
```

Для дополнительных примеров работы с релизами я создам ветку для новой функциональности (рис.29)

```
aiifedorova@aifedorova:~/work/git-extended$ git flow feature start feature_branch
Переключились на новую ветку «vfeature_branch»

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

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

git flow feature finish feature_branch
```

Объединяю ветку feature_branch с develop (рис.30)

```
aiifedorova@aifedorova:~/work/git-extended$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка vfeature_branch удалена (была 761002d).

Summary of actions:
- The feature branch 'vfeature_branch' was merged into 'develop'
- Feature branch 'vfeature_branch' has been locally deleted
- You are now on branch 'develop'
```

Теперь я создаю релиз с версией 1.2.3 (рис.31)

```
aiifedorova@aifedorova:~/work/git-extended$ git flow release start 1.2.3
Переключились на новую ветку «v1.2.3»

Summary of actions:
- A new branch 'v1.2.3' was created, based on 'develop'
- You are now on branch 'v1.2.3'

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'
```

Я снова редактирую файл package.json, меняя текущую версию на 1.2.3. (рис.32)

```
package.json  [-M--] 19 L: [ 1+ 7  8/ 15] *(231 / 342b) 0034 0x022
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "repository": "https://github.com/Angelica23i4i4/git-extended.git",
  "author": "aiifedorova i1igg0000966@gmail.com",
  "license": "1.2.3",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Снова создам журнал изменений (рис.33)

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

Добавлю журнал изменений в индекс (рис.34)

```
aifedorova@aifedorova:~/work/git-extended$ git add CHANGELOG.md
aifedorova@aifedorova:~/work/git-extended$ git commit -am 'chore(site): update changelog'
[v1.2.3 42a0719] chore(site): update changelog
1 file changed, 1 insertion(+), 1 deletion(-)
```

Снова перенаправляю релизную ветку в основную ветку и отвечаю на необходимые вопросы для этого(ярлык v оставлю прежним) (рис.35)

```
Merge tag 'v1.2.3' into develop
v
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной вышестоящей
# ветки в тематическую ветку.
#
# Строки, начинающиеся с «#» будут проигнорированы, а пустое сообщение
# отменяет процесс коммита.
```

Вижу, что процесс завершился без ошибок(рис.36)

```
aifedorova@aifedorova:~/work/git-extended$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
 package.json | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
warning: refname 'v1.2.3' is ambiguous.
warning: refname 'v1.2.3' is ambiguous.
Merge made by the 'ort' strategy.
 package.json | 2 +-

```

Отправляю данные на github (рис.37)

```
aifedorova@aifedorova:~/work/git-extended$ git push --all
Перечисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 581 байт | 581.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/Angelica23i4i4/git-extended.git
 761002d..631ce71 develop -> develop
 b9ceb72..28612b3 main -> main
```






Далее отправляю все теги на github(рис.38)

```
aifedorova@aifedorova:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байтов | 155.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Angelica23i4i4/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
```

Снова создам релиз на github с комментарием из журнала изменений (рис.39)

```
aifedorova@aifedorova:~/work/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Angelica23i4i4/git-extended/releases/tag/v1.2.3
```

Захожу на в свой репозиторий на сайте и вижу, что продленные изменения и созданные файлы успешно лежат в созданном репозитории.(рис.40)

 Angelica23i4i4 'Practice'		28612b3 · 4 minutes ago	 7 Commits
 CHANGELOG.md	chore(site): add changelog	23 minutes ago	
 README.md	first commit	10 hours ago	
 package.json	chore(site): update changelog	4 minutes ago	






Re

5 Выводы

Я получила навыки правильной работы с репозиториями git.

6 Список литературы

Лабораторная работа №4