

Отчёт по лабораторной работе №4

Дисциплина: Операционные системы

Нечаева Кира Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.1.1	<i>Установка git-flow</i>	7
3.1.2	<i>Установка Node.js</i>	8
3.1.3	<i>Настройка Node.js</i>	8
3.1.4	<i>Работа с репозиторием git. Разработка новой функциональности</i>	12
4	Вывод	14
	Список литературы	15

Список иллюстраций

Список таблиц

1 Цель работы

Целью работы является получение навыков правильной работы с репозиториями git.

2 Задание

1. Установка программного обеспечения

1.1. Установка git-flow

1.2. Установка Node.js

1.3. Настройка Node.js

1.3.1 Первоначальная настройка Node.js

1.3.2 Создание репозитория git

1.3.3 Конфигурация для пакетов Node.js

1.4. Работа с репозиторием git. Разработка новой функциональности

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

3.1.1 Установка *git-flow*

Поскольку я работаю на Fedora, то мне потребуется сначала включить репозиторий copr. (рис. [??])

```
[kanechaeva@192 ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для kanechaeva:
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
```

торий copr. (рис. [??])

```
[kanechaeva@192 ~]$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos 5.5 kB/s | 1
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий
=====
Установка:
gitflow    x86_64       1.12.3-1.fc34  copr:copr.fedorainfracloud.org:elegos:gi
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]: y
Загрузка пакетов:
gitflow-1.12.3-1.fc34.x86_64.rpm 457 kB/s |
=====
```

И затем уже я устанавливаю git-flow. (рис. [??])

3.1.2 Установка Node.js

Теперь мне нужно установить Node.js. (рис. [??])

```
[kanechaeva@192 ~]$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:30 назад, Ср 06 мар 2024 21:34
Зависимости разрешены.
=====
Пакет      Архитектура  Версия
-----
Установка:
nodejs     x86_64       1:18.19.0-1.fc38
Установка зависимостей:
nodejs-libs x86_64       1:18.19.0-1.fc38
Установка слабых зависимостей:
nodejs-docs noarch       1:18.19.0-1.fc38
nodejs-full-i18n x86_64       1:18.19.0-1.fc38
nodejs-npm  x86_64       1:10.2.3-1.18.19.0.1.fc38
=====
Результат транзакции
=====
Установка 5 Пакетов
Объем загрузки: 35 М
```

```
[kanechaeva@192 ~]$ sudo npm install -g pnpm
added 1 package in 870ms

1 package is looking for funding
  run `npm fund` for details

npm notice
npm notice New minor version of npm available! 10.2
npm notice Changelog: https://github.com/npm/cli/releases
npm notice Run npm install -g npm@10.5.0 to update!
npm notice
```

И после этого я скачиваю через npm pnpm. (рис. [??])

3.1.3 Настройка Node.js

3.1.3.1 Первоначальная настройка Node.js

Для начала я запускаю pnpm, после выполняю команду для того, чтобы “перелогиниться”. И в конце я использую программу commitizen для помощи в форматировании коммитов. (рис. [??])

```
[kanechaeva@192 ~]$ pnpm setup
No changes to the environment were made. Everything is already up to date.
[kanechaeva@192 ~]$ source ~/.bashrc
[kanechaeva@192 ~]$ ^C
[kanechaeva@192 ~]$ pnpm add -g commitizen
Packages: +152
Progress: resolved 152, reused 0, downloaded 152, added 152, done
Downloading registry.npmjs.org/typescript/5.4.2: 5.82 MB/5.82 MB, done
/home/kanechaeva/.local/share/pnpm/global/5:
+ commitizen 4.3.0
Done in 4.8s
```

Теперь для создания логов я использую следующую команду. (рис. [??])

Теперь для создания логов я использую следующую команду. (рис. [??])


```
[kanechaeva@192 ~]$ pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done

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

Done in 4.4s
```

3.1.3.2 Создание репозитория git

Теперь я создаю репозиторий на гитхаб. (рис. [??])

```
[kanechaeva@192 ~]$ gh repo create git-extended
✓ Created repository kanechaeva/git-extended on GitHub
```

Сразу клонирую его к себе в рабочее пространство на компьютере. (рис. [??])

```
[kanechaeva@192 ~]$ git clone --recursive https://github.com/kanechaeva/git-extended
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
```

После этого я делаю первый коммит и выкладываю на github. (рис. [??])

```
[kanechaeva@192 work]$ cd git-extended
[kanechaeva@192 git-extended]$ git commit -m "first commit"
Текущая ветка: main

Начальный коммит

ничего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
[kanechaeva@192 git-extended]$ git remote add origin git@github.com:kanechaeva/git-extended.git
bash: kanechaeva: Нет такого файла или каталога
[kanechaeva@192 git-extended]$ git remote add origin https://github.com/kanechaeva/git-extended
error: внешний репозиторий origin уже существует
[kanechaeva@192 git-extended]$ ^C
[kanechaeva@192 git-extended]$ git push -u origin master
error: src refspec master не может быть создан
```

3.1.3.3 Конфигурация для пакетов Node.js

Пишу команду `pnpm init`. Вообще необходимо заполнить несколько параметров, но поскольку я иначе скачивала этот пакет, то у меня уже всё заполнено.

```
[kanechaeva@192 git-extended]$ pnpm init
Wrote to /home/kanechaeva/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. [??])

Необходимо

заполнить несколько параметров пакета.

Добавляю файлы, выполняю коммит и отправляю их (файлы) на гитхаб.

```
[kanechaeva@192 git-extended]$ git add .
[kanechaeva@192 git-extended]$ git cz
[main (корневой коммит) 237c152] Текущая ветка: main Начальный коммит
1 file changed, 12 insertions(+)
create mode 100644 package.json
```

(рис. [??]), (рис. [??])

```
[kanechaeva@192 git-extended]$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1.16 КиБ | 1.17 МиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/kanechaeva/git-extended
* [new branch]      main -> main
```

Теперь инициализирую git-flow, префикс для ярлыков я ставлю “v”. (рис. [??])

```
[kanechaeva@192 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/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [/home/kanechaeva/work/git-extended/.git/hooks]
```

Проверяю, что я на ветке develop и загружаю весь репозиторий в хранилище.

```
[kanechaeva@192 git-extended]$ git branch
* develop
  main
[kanechaeva@192 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/kanechaeva/git-extended/pull/new/develop
remote:
To https://github.com/kanechaeva/git-extended
* [new branch]      develop -> develop
```

(рис. [??])

Устанавливаю внешнюю ветку как вышестоящую для этой ветки, создаю релиз с версией 1.0.0 и журнал изменений. Затем добавляю журнал изменений в индекс.

```
[kanechaeva@192 git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[kanechaeva@192 git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «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'

[kanechaeva@192 git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[kanechaeva@192 git-extended]$ ^C
bash: : команда не найдена...
[kanechaeva@192 git-extended]$ git add CHANGELOG.md
[kanechaeva@192 git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 54f0dd7] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
```

(рис. [??])

После всего этого заливаю релизную ветку в основную ветку, отправляю дан-

```
[kanechaeva@192 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».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 54f0dd7).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'

[kanechaeva@192 git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 2.65 КиБ | 2.65 МБ/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно использован
To https://github.com/kanechaeva/git-extended
237c152..dbae3f6d develop -> develop
237c152..287fbbd main -> main
[kanechaeva@192 git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 161 байт | 161.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использован
To https://github.com/kanechaeva/git-extended
* [new tag] v1.0.0 -> v1.0.0
[kanechaeva@192 git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
```

ные на github и там же создаю релиз. (рис. [??])

3.1.4 Работа с репозиторием *git*. Разработка новой функциональности

В начале создаю ветку для новой функциональности. По окончании разработки новой функциональности следующим шагом объединяю ветку `feature_branch` с `develop` и создаю релиз с версией 1.2.3. (рис. [??])

```
[kanechaeva@192 git-extended]$ git flow feature start feature_branch
Переключились на новую ветку «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

[kanechaeva@192 git-extended]$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была dbe3f6d).

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

[kanechaeva@192 git-extended]$ git flow release start 1.2.3
Переключились на новую ветку «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'
```

Теперь создаю журнал изменений и добавляю его в индекс. (рис. [??])

```
[kanechaeva@192 git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[kanechaeva@192 git-extended]$ git add CHANGELOG.md
[kanechaeva@192 git-extended]$ git commit -am 'chore(site): update changelog'
[release/1.2.3 5debf93] chore(site): update changelog
2 files changed, 11 insertions(+), 1 deletion(-)
```

Наконец заливаю релизную ветку в основную ветку и оправаю данные на github.

```

[kanechaeva@192 git-extended]$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 5 +++++
  package.json | 7 +++++-
  2 files changed, 11 insertions(+), 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».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 5 +++++
  package.json | 7 +++++-
  2 files changed, 11 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 5deb93).

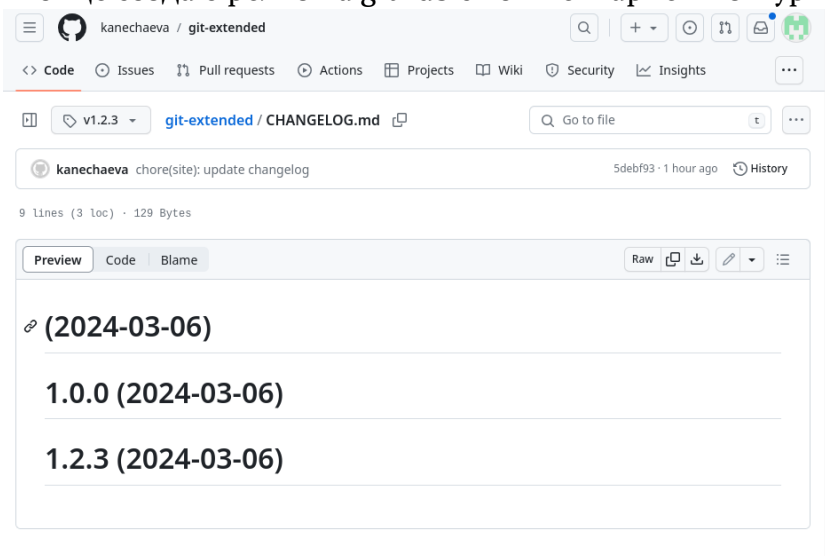
Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

[kanechaeva@192 git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 3.01 КиБ | 3.01 МиБ/с, готово.
Всего 6 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/kanechaeva/git-extended
   db3f6d..5d4df57 develop -> develop
   287fbbd..6202885 main -> main
[kanechaeva@192 git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 160 байтов | 160.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/kanechaeva/git-extended
 * [new tag]           v1.2.3 -> v1.2.3
[kanechaeva@192 git-extended]$ gh release create v1.2.3 -F CHANGELOG.md

```

(рис. [??])

И в конце создаю релиз на github с комментарием из журнала изменений. (рис.



[??])

4 Вывод

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

Список литературы