

Лабораторная работа № 4

Операционные системы

Клименко Алёна Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Рабочий процесс Gitflow	7
4	Выполнение лабораторной работы	8
4.1	Установка программного обеспечения.	8
4.1.1	Установка git-flow	8
4.1.2	Установка и настройка node.js	8
4.1.3	Общепринятые коммиты.	9
4.2	Практический сценарий использования git.	10
4.2.1	Создание репозитория git.	10
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Установка gitflow в режиме суперпользователя	8
4.2	Установка приложений	9
4.3	Настройка node.js	9
4.4	Настройка программ	10
4.5	Создание репозитория, первый коммит	11
4.6	git cz	11
4.7	Изменения файла	12
4.8	Выполнение коммита	12
4.9	Команда push	13

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

3.1 Рабочий процесс Gitflow

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow. Общая информация 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

4.1 Установка программного обеспечения.

4.1.1 Установка git-flow

1. Открываем терминал и входим в режим суперпользователя, устанавливаем gitflow. (рис. 4.1).

```
[asklinenko@asklinenko ~]$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/develop/contrib/gitflow-installer.sh
[asklinenko@asklinenko ~]$ ls
config  gitflow-installer.sh  work  Видео  Документы  Загрузки  Избранное  Музыка  Общедоступные  Рабочий стол  Шаблоны
[asklinenko@asklinenko ~]$ sudo bash gitflow-installer.sh install stable
## git-flow no-make installer ##
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование с «gitflow»...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270 (from 1)
Получение объектов: 100% (4270/4270), 1.74 Мб | 3.92 Мб/с, готово.
Определение изменений: 100% (2533/2533), готово.
Уже актуально.
branch 'master' set up to track 'origin/master'.
Переключились на новую ветку «master»
install: создание каталога '/usr/local/share/doc'
install: создание каталога '/usr/local/share/doc/gitflow'
install: создание каталога '/usr/local/share/doc/gitflow/hooks'
'gitflow/git-flow' -> '/usr/local/bin/git-flow'
'gitflow/git-flow-init' -> '/usr/local/bin/git-flow-init'
'gitflow/git-flow-feature' -> '/usr/local/bin/git-flow-feature'
'gitflow/git-flow-bugfix' -> '/usr/local/bin/git-flow-bugfix'
'gitflow/git-flow-hotfix' -> '/usr/local/bin/git-flow-hotfix'
'gitflow/git-flow-release' -> '/usr/local/bin/git-flow-release'
'gitflow/git-flow-support' -> '/usr/local/bin/git-flow-support'
'gitflow/git-flow-version' -> '/usr/local/bin/git-flow-version'
'gitflow/gitflow-common' -> '/usr/local/bin/gitflow-common'
'gitflow/gitflow-shFlags' -> '/usr/local/bin/gitflow-shFlags'
'gitflow/git-flow-config' -> '/usr/local/bin/git-flow-config'
'gitflow/hooks/filter-flow-hotfix-finish-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-hotfix-finish-tag-message'
'gitflow/hooks/filter-flow-hotfix-start-version' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-hotfix-start-version'
'gitflow/hooks/filter-flow-release-branch-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-branch-tag-message'
'gitflow/hooks/filter-flow-release-finish-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-finish-tag-message'
'gitflow/hooks/filter-flow-release-start-version' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-start-version'
```

Рис. 4.1: Установка gitflow в режиме суперпользователя

4.1.2 Установка и настройка node.js

2. Устанавливаем npm и nodejs. (рис. 4.2).


```
asklinenko@asklinenko ~$ sudo dnf install nodejs prpm
[sudo] пароль для asklinenko:
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates
Репозитории загружены.
Пакет
Установка:
nodejs
prpm
Установка зависимостей:
nodejs-libs
Установка слабых зависимостей:
nodejs-docs
nodejs-full-libs
nodejs-prpm
Сводная транзакция:
Установка: 6 пакетов

Аrch. Версия
x86_64 1:22.14.0-2.fc41
noarch 9.13.0-1.fc41
x86_64 1:22.14.0-2.fc41
noarch 1:22.14.0-2.fc41
x86_64 1:22.14.0-2.fc41
x86_64 1:10.9.2-1.22.14.0-2.fc41

Репозиторий
updates
updates
updates
updates
updates
updates

Размер
149.5 KiB
15.5 MiB
75.9 MiB
93.7 MiB
30.4 MiB
9.3 MiB

Общий размер входящих пакетов составляет 42 MiB. Необходимо загрузить 42 MiB.
После этой операции будут использоваться дополнительные 225 MiB (установка 225 MiB, удаление 0 B).
[1/6] nodejs-1:22.14.0-2.fc41.x86_64
[2/6] prpm-9.13.0-1.fc41.noarch
[3/6] nodejs-prpm-1:10.9.2-1.22.14.0-2.fc41.x86_64
[4/6] nodejs-docs-1:22.14.0-2.fc41.noarch
[5/6] nodejs-full-libs-1:22.14.0-2.fc41.x86_64
[6/6] nodejs-libs-1:22.14.0-2.fc41.x86_64
-----
[6/6] Total
100% | 37.8 KiB/s | 25.6 KiB | 00m01s

Выполнение транзакции
[1/8] Проверить файлы пакета
[2/8] Подготовить транзакцию
[3/8] Установка nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/8] Установка nodejs-1:22.14.0-2.fc41.x86_64
[5/8] Установка prpm-9.13.0-1.fc41.noarch
[6/8] Установка nodejs-prpm-1:10.9.2-1.22.14.0-2.fc41.x86_64
[7/8] Установка nodejs-full-libs-1:22.14.0-2.fc41.x86_64
[8/8] Установка nodejs-docs-1:22.14.0-2.fc41.noarch
Завершено!
asklinenko@asklinenko ~$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/develop/contrib/gitflow-installer.sh
asklinenko@asklinenko ~$ ls
config gitflow-installer.sh work
asklinenko@asklinenko ~$ sudo bash gitflow-installer.sh install stable
asklinenko@asklinenko ~$
```

Рис. 4.2: Установка приложений

3. Настраиваем nodejs. (рис. 4.3).

```
bash: prpm: команда не найдена
asklinenko@asklinenko ~$ prpm setup
Appended new lines to /home/asklinenko/.bashrc

Next configuration changes were made:
export PNP_HOME="/home/asklinenko/.local/share/prpm"
case "$PATH" in
  *) export PNP_HOME="$PNP_HOME" ;;
  *) export PATH="$PNP_HOME:$PATH" ;;
esac

To start using prpm, run:
source /home/asklinenko/.bashrc
asklinenko@asklinenko ~$
```

Рис. 4.3: Настройка node.js

4.1.3 Общепринятые коммиты.

4. Настраиваем commitizen и standard-changelog (рис. 4.4).

```
To start using prpm, run:
source /home/asklinenko/.bashrc
[asklinenko@asklinenko ~]$ source .bashrc
[asklinenko@asklinenko ~]$ prpm add -g commitizen

Update available! 9.13.0 - 10.6.0.
Changelog: https://github.com/prpm/prpm/releases/tag/v10.6.0
Run "prpm self-update" to update.
Follow @prpmjs for updates: https://x.com/prpmjs

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/asklinenko/.local/share/prpm/global/5:
commitizen @ 3.1.1

Done in 6.5s
[asklinenko@asklinenko ~]$ prpm 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

/home/asklinenko/.local/share/prpm/global/5:
standard-changelog @ 0.0.0

Done in 4.1s
[asklinenko@asklinenko ~]$ gh repo create --public git-extended
Created repository Alstrr/git-extended on GitHub
https://github.com/Alstrr/git-extended
[asklinenko@asklinenko ~]$
```

Рис. 4.4: Настройка программ

4.2 Практический сценарий использования git.

4.2.1 Создание репозитория git.

5. Создаем репозиторий git, настраиваем его и делаем в него первый коммит. Настраиваем пакет файлов nodejs (рис. 4.5).

```
Alstir/git-extended at feature/fi[Browser] | foot
create mode 100644 package.json
[asklinenko@asklinenko git-extended]$ git push
Перечисление объектов: 4, готово.
Пакет объектов: 100% (4/4), готово.
Сжать объекты: 100% (3/3), готово.
Запись объектов: 100% (3/3), 510 байт | 510.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Alstir/git-extended
 0154c53..37acb60 main -> main
[asklinenko@asklinenko git-extended]$ git flow init
Fatal: Working tree contains unstaged changes. Aborting.
[asklinenko@asklinenko git-extended]$ git flow init
Fatal: Working tree contains unstaged changes. Aborting.
[asklinenko@asklinenko git-extended]$ git add
[asklinenko@asklinenko git-extended]$ git push
Everything up-to-date
[asklinenko@asklinenko git-extended]$ git flow init
Fatal: Index contains uncommitted changes. Aborting.
[asklinenko@asklinenko git-extended]$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
изменено:   package.json

[asklinenko@asklinenko git-extended]$ git commit -m 'try'
(main 2b3213f) try
1 file changed, 2 insertions(+), 1 deletion(-)
[asklinenko@asklinenko git-extended]$ git flow init

Which branch should be used for bringing forth production releases?
  main
Branch name for production releases: [main]
Branch name for "next release" development: [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/asklinenko/git-extended/.git/hooks]
```

Рис. 4.5: Создание репозитория, первый коммит

используем git cz (рис. 4.6).

```
1 | foot | 40% ◀ 100% ▾ | 10.0.2.15/24 | 3% 🔌 45% ☰ 10% | 00:45
Alstir/git-extended at feature/fi[Browser] | foot
рекурсивно по подмодулям
--[no-]format <формат>
      |
      | использовать формат для вывода
[asklinenko@asklinenko git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[asklinenko@asklinenko 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'

[asklinenko@asklinenko git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[asklinenko@asklinenko git-extended]$ git add .
[asklinenko@asklinenko git-extended]$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

Select the type of change that you're committing: chore:   Other changes that don't modify src or test files
What is the scope of this change (e.g. component or file name): (press enter to skip) site
Write a short, imperative tense description of the change (max 87 chars):
[!] add changelog
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
[release/1.0.0 7d31df7] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
[asklinenko@asklinenko 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(+)
```

Рис. 4.6: git cz

вфплняем коммиты создаем новую ветку проверяем через git branch и делаем релиз (рис. 4.7).

```
1 Alstir/git-extended at feature/f(Browser) | foot
[asklinenko@asklinenko git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 555 байт | 555.00 Кб/с, готово.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:Alstir/git-extended
  2b3213f..075470e develop -> develop
  2b3213f..8a68bd9 main -> main
[asklinenko@asklinenko git-extended]$ git push --all
Everything up-to-date
[asklinenko@asklinenko git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 153 байт | 153.00 Кб/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Alstir/git-extended
  * [new tag] v1.0.0 -> v1.0.0
[asklinenko@asklinenko git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Alstir/git-extended/releases/tag/v1.0.0
[asklinenko@asklinenko git-extended]$ git flow feature start feature_branch
Переключимся 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

[asklinenko@asklinenko git-extended]$ git branch
  develop
* feature/feature_branch
  main
[asklinenko@asklinenko git-extended]$ git flow release start 1.2.3
Переключимся 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:
```

Рис. 4.7: Изменения файла

7. Выполняем коммит (рис. 4.8). Выкладываем на github. (рис. 4.9).

```
1 Alstir/git-extended at feature/f(Browser) | foot
git flow release finish '1.2.3'

[asklinenko@asklinenko git-extended]$ nano package.json
[asklinenko@asklinenko git-extended]$ satndard-changelog
bash: satndard-changelog: команда не найдена
[asklinenko@asklinenko git-extended]$ standard-changelog
+ output changes to CHANGELOG.md
[asklinenko@asklinenko git-extended]$ git add .
[asklinenko@asklinenko git-extended]$ git status
Текущая ветка: release/1.2.3
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   CHANGELOG.md
    изменено:   package.json

[asklinenko@asklinenko git-extended]$ git cz
cz-cl@04.3.1, cz-conventional-changelog@3.3.0

Select the type of change that you're committing: chore - Other changes that don't modify src or test files
What is the scope of this change (e.g. component or file name): (press enter to skip) site
Write a short, imperative tense description of the change (max 87 chars):
(13) upd changelog
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
[release/1.2.3 40fdee0] chore(site): upd changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[asklinenko@asklinenko git-extended]$ git flow release finish 1.2.3
Переключимся на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Уво на «main»
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключимся на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 4.8: Выполнение коммита

```

CHANGELOG.md | 4 +++
package.json | 2 +-
2 files changed, 5 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 | 4 +++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 40fdce0).

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'

[asklimenko@asklimenko git-extended]$ git push -all
error: имелось в виду '--all' (с двумя дефисами)?
[asklimenko@asklimenko git-extended]$ git push --al
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 744 байта | 744.00 Киб/с, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:Alstrr/git-extended
   075470e..b3ac583  develop -> develop
   8a68bd9..5e53e94  main -> main
* [new branch]      feature/feature_branch -> feature/feature_branch
[asklimenko@asklimenko git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 152 байта | 152.00 Киб/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Alstrr/git-extended
* [new tag]         v1.2.3 -> v1.2.3
[asklimenko@asklimenko git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Alstrr/git-extended/releases/tag/v1.2.3
[asklimenko@asklimenko git-extended]$

```

Рис. 4.9: Команда push

5 Выводы

В процессе выполнения лабораторной работы я приобрела навыки правильной работы с репозиториями git.

Список литературы