

Лабораторная работа № 4

Отчет

Казначеев Сергей Ильич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Вывод	21
	Список литературы	22

Список иллюстраций

4.1	sc1	8
4.2	sc2	9
4.3	sc3	9
4.4	sc4	10
4.5	sc5	10
4.6	sc6	11
4.7	sc7	11
4.8	sc8	12
4.9	sc9	12
4.10	sc10	12
4.11	sc11	12
4.12	sc12	13
4.13	sc13	13
4.14	sc14	14
4.15	sc15	15
4.16	sc16	15
4.17	sc17	15
4.18	sc18	15
4.19	sc19	16
4.20	sc21	16
4.21	sc22	17
4.22	sc23	17
4.23	sc24	17
4.24	sc25	18
4.25	sc26	18
4.26	sc27	19
4.27	sc28	20

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

Выполнить работу для тестового репозитория. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

Для начала подключим репозиторий из которого можно скачать gitflow

```
[iskaznacheev@iskaznacheev ~]$ dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rp 100% | 668.0  B/s | 652.0  B | 00m01s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является част
ью
основного дистрибутива, и его качество может отличаться.

Проект Fedora не имеет никакой власти над содержимым
за пределами правил, изложенных в Copr FAQ по адресу
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>
и пакеты не имеют никаких требований к качеству или уровню безопасности.

Пожалуйста, не публикуйте сообщения об ошибках, связанных с этими пакетами, в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.
Is this ok [y/N]:
```

Рис. 4.1: sc1

После установим gitflow


```

foot
[iskaznacheev@iskaznacheev ~]$ sudo dnf install gitflow
Обновление и загрузка репозитория:
  Copr repo for gitflow owned by elegos      100% | 2.9 KiB/s | 2.4 KiB | 00m01s
Репозитории загружены.
Пакет      Арх.  Версия      Репозиторий      Размер
Установка:
  gitflow   x86_64 1:1.12.3-1.fc34  copr:copr.fedora 261.7 KiB
Сводка транзакции:
  Установка:      1 пакета
Общий размер входящих пакетов составляет 57 KiB. Необходимо загрузить 57 KiB.
После этой операции будут использоваться дополнительные 262 KiB (установка 262 KiB,
удаление 0 B).
Is this ok [y/N]: y
[1/1] gitflow-0:1.12.3-1.fc34.x86_64      100% | 339.8 KiB/s | 57.4 KiB | 00m01s
-----
Total: 100% | 339.8 KiB/s | 57.4 KiB | 00m01s

```

Рис. 4.2: sc2

Теперь установим Nodejs

```

[iskaznacheev@iskaznacheev ~]$ sudo dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет      Арх.  Версия      Репозиторий
Установка:
  nodejs    x86_64 1:22.14.0-2.fc41  updates  141
Установка зависимостей:
  nodejs-libs  x86_64 1:22.14.0-2.fc41  updates  75
Установка слабых зависимостей:
  nodejs-docs  noarch 1:22.14.0-2.fc41  updates  90
  nodejs-full-i18n  x86_64 1:22.14.0-2.fc41  updates  30
  nodejs-ppm    x86_64 1:10.9.2-1.22.14.0.2.fc41  updates  1
Сводка транзакции:
  Установка:      5 пакетов

```

Рис. 4.3: sc3

Установим ppm

```
[iskaznacheev@iskaznacheev ~]$ 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

Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 3 MiB. Необходимо загрузить 3 MiB.
После этой операции будут использоваться дополнительные 16 MiB (установка 16 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] pnpm-0:9.13.0-1.fc41.noarch 100% | 7.7 MiB/s | 3.0 MiB | 00m00s
-----
[1/1] Total 100% | 3.0 MiB/s | 3.0 MiB | 00m00s
Выполнение транзакции
[1/3] Проверить файлы пакетов 100% | 37.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакцию 100% | 6.0 B/s | 1.0 B | 00m00s
[3/3] Установка pnpm-0:9.13.0-1.fc41.noarch 100% | 18.0 MiB/s | 15.5 MiB | 00m01s
```

Рис. 4.4: sc4

Запустим pnpm

```
[iskaznacheev@iskaznacheev ~]$ pnpm setup
Appended new lines to /home/iskaznacheev/.bashrc

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

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

Рис. 4.5: sc5

Теперь установим с помощью него Commitizen

```

[iskaznacheev@iskaznacheev ~]$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 151, reused 151, downloaded 0, added 0, done
Done in 1.6s
[iskaznacheev@iskaznacheev ~]$ 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

/home/iskaznacheev/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

```

Рис. 4.6: sc6

Создаем тестовый репозиторий git-extended

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *

Kava-45 ▾

Repository name *

git-extended

✓ git-extended is available.

Great repository names are short and memorable. Need inspiration?
How about **congenial-happiness** ?

Рис. 4.7: sc7

И клонируем его себе на компьютер

```
[iskaznacheev@iskaznacheev ~]$ git clone --recursive git@github.com:Kava-45/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
```

Рис. 4.8: sc8

Создадим какойнибудь файл и проиндексируем его с помощью `git add` сделав при этом `commit`

```
[iskaznacheev@iskaznacheev git-extended]$ touch README.md
[iskaznacheev@iskaznacheev git-extended]$ git add .
[iskaznacheev@iskaznacheev git-extended]$ git commit -m "first commit"
[main (корневой коммит) c0d9c2f] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
```

Рис. 4.9: sc9

Теперь добавим ветку

```
[iskaznacheev@iskaznacheev git-extended]$ git remote add origin git@github.com:Kava-45/git-extended.git
error: внешний репозиторий origin уже существует
[iskaznacheev@iskaznacheev git-extended]$
```

Рис. 4.10: sc10

Теперь обратно запустим на github

```
[iskaznacheev@iskaznacheev git-extended]$ git push -u origin main
```

Рис. 4.11: sc11

Теперь проинициализируем `pnpm`

```
[iskaznacheev@iskaznacheev git-extended]$ npm init
Wrote to /home/iskaznacheev/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. 4.12: sc12

После инициализации создается файл package.json который меняем следующим образом

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Kava-45/git-extended.git",
  "keywords": [],
  "author": "Sergey Kaznacheev mechtitana@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.13: sc13

Делаем коммит с помощью cz

```

[iskaznacheev@iskaznacheev 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) readme.md
? Write a short, imperative tense description of the change (max 83 chars):
  (10) added file
? 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 b0466ea] feat(readme.md): added file
 1 file changed, 15 insertions(+)
  create mode 100644 package.json
[iskaznacheev@iskaznacheev git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.17 КиБ | 1.17 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Kava-45/git-extended.git
   c0d9c2f..b0466ea  main -> main
[iskaznacheev@iskaznacheev git-extended]$ █

```

Рис. 4.14: sc14

Проинициализируем gitflow. Укажем название веток и префикс для версий

```

[iskaznacheev@iskaznacheev git-extended]$ git flow init -f

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

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

```

Рис. 4.15: sc15

Выведем список веток и убедимся, что мы находимся в develop, и запустим изменения на сервер

```

[iskaznacheev@iskaznacheev git-extended]$ git branch
* develop
  main

```

Рис. 4.16: sc16

Переключимся на ветку develop после чего создадим ветку релиза где создадим changelog

```

[iskaznacheev@iskaznacheev git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 4.17: sc17

```

[iskaznacheev@iskaznacheev git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md

```

Рис. 4.18: sc18

Проиндексируем changelog и сделаем коммит

```
iskaznacheev@iskaznacheev git-extended]$ git add CHANGELOG.md
iskaznacheev@iskaznacheev git-extended]$ git commit -am 'chore(site): add changelog'

[release/1.0.0 4b4c605] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
iskaznacheev@iskaznacheev git-extended]$
```

Рис. 4.19: sc19

Теперь сольем ветку release с веткой changelog

```
iskaznacheev@iskaznacheev 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'

iskaznacheev@iskaznacheev git-extended]$
```

Рис. 4.20: sc21

Загружаем данные на github


```
[iskaznacheev@iskaznacheev git-extended]$ git push --all
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 892 байта | 892.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Kava-45/git-extended.git
    b0466ea..87615cd  develop -> develop
[iskaznacheev@iskaznacheev git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 150 байтов | 150.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Kava-45/git-extended.git
    * [new tag]          v1.0.0 -> v1.0.0
[iskaznacheev@iskaznacheev git-extended]$
```

Рис. 4.21: sc22

Создаем релиз из changelog'a

```
[iskaznacheev@iskaznacheev git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Kava-45/git-extended/releases/tag/v1.0.0
[iskaznacheev@iskaznacheev git-extended]$
```

Рис. 4.22: sc23

Создаем ветку feature и сразу сливаем ее с develop

```
[iskaznacheev@iskaznacheev 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
```

Рис. 4.23: sc24

Создаем ветку релиза

```
[iskaznacheev@iskaznacheev 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'
```

Рис. 4.24: sc25

Меняем версию в package.json

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Kava-45/git-extended.git",
  "keywords": [],
  "author": "Sergey Kaznacheev mechtitana@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.25: sc26

Создаем ветку журнал изменений, проиндексируем его и сольем ветку с ним в ветку develop

```

[iskaznacheev@iskaznacheev git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[iskaznacheev@iskaznacheev git-extended]$ git add CHANGELOG.md
[iskaznacheev@iskaznacheev git-extended]$ git commit -am 'chore(site): update changelog'
[release/1.2.3 @f472cf] chore(site): update changelog
 2 files changed, 5 insertions(+), 1 deletion(-)
[iskaznacheev@iskaznacheev 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 | 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 @f472cf).

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'

```

Рис. 4.26: sc27

Загрузим изменения на гитхаб и создадим релиз

```

[iskaznacheev@iskaznacheev git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.75 КиБ | 1.38 МБ/с, готово.
Total 6 (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:Kava-45/git-extended.git
   87615cd..12e05e4  develop -> develop
   e5a7dbb..93231a4  main -> main
[iskaznacheev@iskaznacheev git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 151 байт | 151.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Kava-45/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
[iskaznacheev@iskaznacheev git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Kava-45/git-extended/releases/tag/v1.2.3
[iskaznacheev@iskaznacheev git-extended]$

```

Рис. 4.27: sc28

5 Вывод

Я получил правильные навыки работы, с репозиториями git

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.