

Отчёт по лабораторной работе 4

Операционные системы

Кижваткина Анна Юрьевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	17

Список иллюстраций

2.1	Консоль	6
2.2	Установка git-flow	7
2.3	Установка Node.js	7
2.4	Установка Node.js	7
2.5	Настройка Node.js	8
2.6	Настройка Node.js	8
2.7	Установка программы	8
2.8	Установка программы	8
2.9	Создание репозитория	9
2.10	Первый коммит	9
2.11	Конфигурация Node.js	10
2.12	Добавление команды для формирования коммитов	10
2.13	Добавление новых файлов	10
2.14	Выполнение коммита	11
2.15	Отправка на github	11
2.16	Инициализация git-flow	11
2.17	Проверка ветки	11
2.18	Загрузка репозитория в хранилище	12
2.19	Установка внешней ветки	12
2.20	Создание релиза	12
2.21	Создание журнала изменений	12
2.22	Добавление журнала изменений	13
2.23	Добавление релизной ветки	13
2.24	Отправка данных	13
2.25	Создание релиза	13
2.26	Создание новой ветки и объединение со старой	14
2.27	Создание релиза	14
2.28	Обновление номера версии	15
2.29	Создание журнала изменений	15
2.30	Добавление журнала изменений	15
2.31	Добавление релизной ветки в основную	15
2.32	Отправка данных на GitHub	16
2.33	Создание релиза	16

Список таблиц

1 Цель работы

Получение практических навыков правильной работы с репозиторием git.

2 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.1).

Заходим в виртуальную машину и открываем консоль с помощью комбинации Win+Enter. (рис. 2.1)

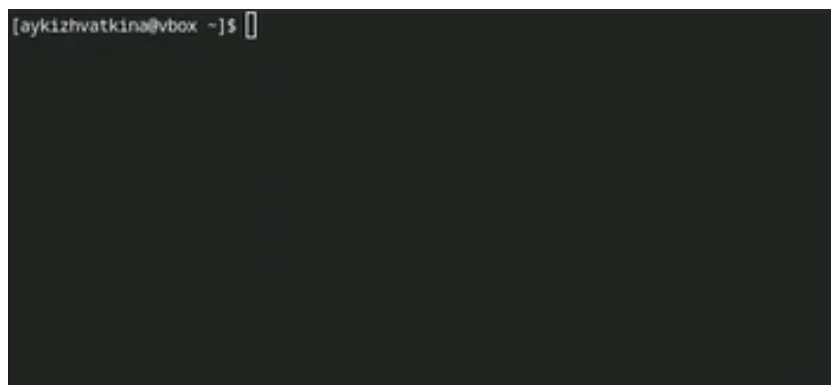


Рис. 2.1: Консоль

Устанавливаем git-flow. (рис. 2.2)

```
[aykizhvatkina@vbox ~]$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для aykizhvatkina:
https://copr.fedorainfracloud.org/api_3/rp 100% | 1.1 KiB/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 содержанием
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]: y
[aykizhvatkina@vbox ~]$ sudo dnf install gitflow
Updating and loading repositories:
Fedora 41 - x86_64 - 777% [<=> ] | 0.0 B/s | 0.0 B | 00m00s
```

Рис. 2.2: Установка git-flow

Устанавливаем Node.js. (рис. 2.3 рис. 2.4)

```
[aykizhvatkina@vbox ~]$ sudo dnf install nodejs
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Upgrading:
libuv x86_64 1:1.50.0-1.fc41 updates 568.4 KiB
replacing libuv x86_64 1:1.49.0-1.fc41 anaconda 563.6 KiB
Installing:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
Installing dependencies:
nodejs-libs x86_64 1:22.14.0-2.fc41 updates 75.9 MiB
Installing weak dependencies:
nodejs-docs noarch 1:22.14.0-2.fc41 updates 93.7 MiB
nodejs-full-i18n x86_64 1:22.14.0-2.fc41 updates 30.4 MiB
nodejs-npm x86_64 1:10.9.2-1.22.14.0.2.fc41 updates 9.3 MiB
```

Рис. 2.3: Установка Node.js

```
[aykizhvatkina@vbox ~]$ sudo dnf install npm
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Installing:
npm noarch 9.13.0-1.fc41 updates 15.5 MiB
Transaction Summary:
Installing: 1 package
```

Рис. 2.4: Установка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавлива-

ем yarn, в переменную path. (рис. 2.5 рис. 2.6)

```
[aykizhvatkina@vbox ~]$ pnpm setup
```

Рис. 2.5: Настройка Node.js

```
[aykizhvatkina@vbox ~]$ source ~/.bashrc
```

Рис. 2.6: Настройка Node.js

Устанавливаем программу для помощи в форматировании коммитов. (рис. 2.7)

```
[aykizhvatkina@vbox ~]$ pnpm add -g commitizen
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/aykizhvatkina/.local/share/pnpm/global/5:
+ commitizen 4.3.1
Done in 7s using pnpm v10.5.2
```

Рис. 2.7: Установка программы

Устанавливаем Standard-changelog для помощи в создании логов. (рис. 2.8)

```
[aykizhvatkina@vbox ~]$ 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/aykizhvatkina/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0
Done in 3.5s using pnpm v10.5.2
```

Рис. 2.8: Установка программы

Создаем репозиторий на GitHub. Называем его git-extended. (рис. 2.9)

Owner *

Anyakizh

Repository name *

git-extended

✓ git-extended is available.

Great repository names are short and memorable. Need inspiration?
How about **effective-goggles** ?

Description (optional)

Рис. 2.9: Создание репозитория

Делаем первый коммит и выкладываем на github. (рис. 2.10)

```
[aykizhvatkina@vbox git-extended]$ git commit -m "first commit"
Текущая ветка: main
Связанная ветка «origin/main» отсутствует в вышестоящем репозитории.
(для исправления запустите «git branch --unset-upstream»)

ничего коммитить, нет изменений в рабочем каталоге
[aykizhvatkina@vbox git-extended]$ git remote add origin git@github.com:Anyakizh/git-extended.git
error: внешний репозиторий origin уже существует
[aykizhvatkina@vbox git-extended]$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:Anyakizh/git-extended.git»
[aykizhvatkina@vbox git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 865 байтов | 865.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anyakizh/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 2.10: Первый коммит

Делаем конфигурацию для пакетов Node.js. Необходимо заполнить несколько параметров пакета. (рис. 2.11)

```
[aykizhvatkina@vbox git-extended]$ npm init
Wrote to /home/aykizhvatkina/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",
  "packageManager": "npm@10.5.2"
}
```

Рис. 2.11: Конфигурация Node.js

Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов. (рис. 2.12)

```
foot
GNU nano 8.1 package.json Изменён
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Anyakizh/git-extended.git",
  "author": "Anna Kizhvatkina annakizvatkina974@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 2.12: Добавление команды для формирования коммитов

Добавим новые файлы. (рис. 2.13)

```
[aykizhvatkina@vbox git-extended]$ git add .
```

Рис. 2.13: Добавление новых файлов

Выполняем коммит. (рис. 2.14)

```
[aykizhvatkina@vbox git-extended]$ git cz
```

Рис. 2.14: Выполнение коммита

Отправляем на github. (рис. 2.15)

```
[aykizhvatkina@vbox git-extended]$ git push
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 1.99 КиБ | 1018.00 КиБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anyakizh/git-extended.git
 a2284ea..eec474d  main -> main
```

Рис. 2.15: Отправка на github

Переходим к конфигурации git-flow. Инициализируем git-flow. (рис. 2.16)

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

Рис. 2.16: Инициализация git-flow

Префикс для ярлыков установим в v.

Проверяем, что мы на ветке develop. (рис. 2.17)

```
[aykizhvatkina@vbox git-extended]$ git branch
* develop
  main
```

Рис. 2.17: Проверка ветки

Загружаем весь репозиторий в хранилище. (рис. 2.18)

```
[aykizhvatkina@vbox 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/Anyakizh/git-extended/pull/new/develop
remote:
To github.com:Anyakizh/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 2.18: Загрузка репозитория в хранилище

Устанавливаем внешнюю ветку как вышестоящую для этой ветки. (рис. 2.19)

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

Рис. 2.19: Установка внешней ветки

Создадим релиз с версией 1.0.0. (рис. 2.20)

```
[aykizhvatkina@vbox 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'
```

Рис. 2.20: Создание релиза

Создадим журнал изменений. (рис. 2.21)

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

Рис. 2.21: Создание журнала изменений

Добавим журнал изменений в индекс. (рис. 2.22)

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

Рис. 2.22: Добавление журнала изменений

Зальём релизную ветку в основную ветку. (рис. 2.23)

```
[aykizhvatkina@vbox git-extended]$ git flow release finish 1.0.0
```

Рис. 2.23: Добавление релизной ветки

Отправим данные на GitHub. (рис. 2.24)

```
[aykizhvatkina@vbox 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), 2.65 КиБ | 2.65 МБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anyakizh/git-extended.git
   eec474d..26c1d1c develop -> develop
   eec474d..198eec0 main -> main
[aykizhvatkina@vbox git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 162 байта | 81.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anyakizh/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
```

Рис. 2.24: Отправка данных

Создадим релиз на GitHub. Для этого используем утилиты работы с Github. (рис. 2.25)

```
[aykizhvatkina@vbox git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
```

Рис. 2.25: Создание релиза

Создадим ветку для новой функциональности. Объединим созданную ветку с develop. (рис. 2.26)

```
[aykizhvatkina@vbox 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

[aykizhvatkina@vbox git-extended]$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка vfeature_branch удалена (была 26c1d1c).

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'
```

Рис. 2.26: Создание новой ветки и объединение со старой

Перейдем к созданию релиза git-flow. Создадим релиз с версией 1.2.3. (рис. 2.27)

```
[aykizhvatkina@vbox git-extended]$ git flow release start 1.2.3
Переключились на новую ветку «v1.2.3»

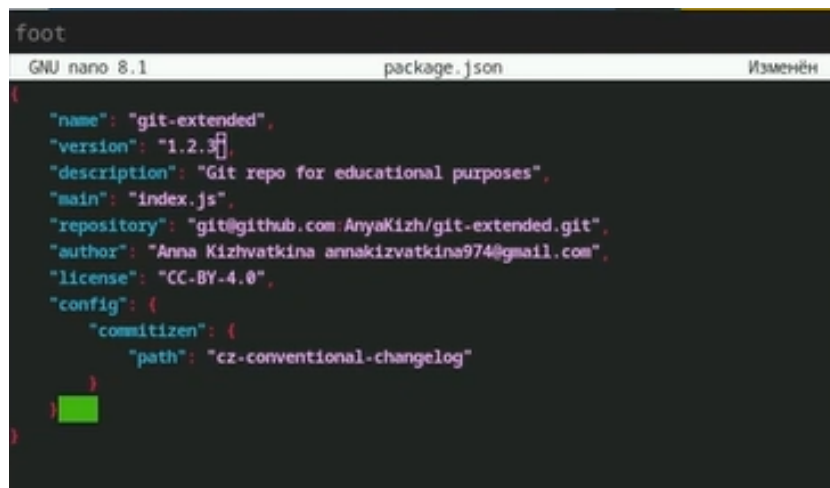
Summary of actions:
- A new branch 'v1.2.3' was created, based on 'develop'
- You are now on branch 'v1.2.3'

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'
```

Рис. 2.27: Создание релиза

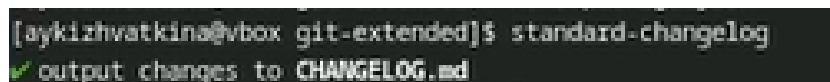
Обновим номер версии в файле package.json. Установим её в 1.2.3. (рис. 2.28)



```
foot
GNU nano 8.1 package.json Изменён
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Anyakizh/git-extended.git",
  "author": "Anna Kizhvatkina annakizvatkina974@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 2.28: Обновление номера версии

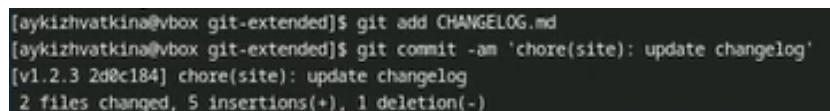
Создадим журнал изменений. (рис. 2.29)



```
[aykizhvatkina@vbox git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
```

Рис. 2.29: Создание журнала изменений

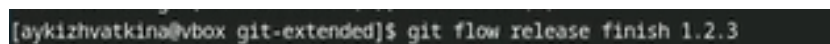
Добавим журнал изменений в индекс. (рис. 2.30)



```
[aykizhvatkina@vbox git-extended]$ git add CHANGELOG.md
[aykizhvatkina@vbox git-extended]$ git commit -am 'chore(site): update changelog'
[v1.2.3 2d0c184] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 2.30: Добавление журнала изменений

Зальём релизную ветку в основную ветку. (рис. 2.31)



```
[aykizhvatkina@vbox git-extended]$ git flow release finish 1.2.3
```

Рис. 2.31: Добавление релизной ветки в основную

Отправляем данные на GitHub. (рис. 2.32)


```

[aykizhvatkina@vbox git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.76 КиБ | 2.76 МБ/с, готово.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Anyakizh/git-extended.git
   26c1d1c..5e5184b  develop -> develop
   198eec0..9b291b6  main -> main
[aykizhvatkina@vbox git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 164 байта | 164.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anyakizh/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3

```

Рис. 2.32: Отправка данных на GitHub

Создадим релиз на GitHub с комментарием из журнала изменений. (рис. 2.33)

```

[aykizhvatkina@vbox git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Anyakizh/git-extended/releases/tag/v1.2.3

```

Рис. 2.33: Создание релиза

3 Выводы

В ходе лабораторной работы я получила навыки правильной работы с репозиторием git.