

Лабораторная работа №4

Операционные системы

Наговицын Арсений Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Установка программного обеспечения.	8
4.2	Настройка общепринятых коммитов.	9
5	Выводы	21

Список иллюстраций

4.1	Окно консоли	8
4.2	Окно консоли	8
4.3	Окно консоли	9
4.4	Окно консоли	9
4.5	Окно консоли	9
4.6	Окно консоли	10
4.7	Окно консоли	10
4.8	Окно консоли	10
4.9	Окно консоли	11
4.10	Окно консоли	11
4.11	Окно консоли	11
4.12	Окно консоли	12
4.13	Окно текстового редактора	12
4.14	Окно консоли	12
4.15	Окно консоли	13
4.16	Окно консоли	13
4.17	Окно консоли	14
4.18	Окно консоли	14
4.19	Окно GNU nano	14
4.20	Окно GNU nano	15
4.21	Окно GNU nano	15
4.22	Окно консоли	16
4.23	Окно консоли	16
4.24	Окно консоли	17
4.25	Окно консоли	17
4.26	Окно текстового редактора	18
4.27	Окно консоли	18
4.28	Окно GNU nano	18
4.29	Окно GNU nano	19
4.30	Окно GNU nano	19
4.31	Окно консоли	19
4.32	Страница сайта github	20

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

1. Установка и настройка программного обеспечения.
2. Настройка общепринятых коммитов.

3 Теоретическое введение

Рабочий процесс Gitflow

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета gitflow.

Общая информация

Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. Создадим ветку Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпусков. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок. 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.

Если в master обнаружена проблема, из master создаётся ветка hotfix.

Когда работа над веткой исправления hotfix завершена, она сливается в ветки master и develop.

4 Выполнение лабораторной работы

4.1 Установка программного обеспечения.

Включаю корпоративный репозиторий командой(рис. 4.1).

```
[avnagovicihn@avnagovicihn ~]$ sudo -i
[root@avnagovicihn ~]# dnf copr enable elegos/gitflow
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.
[root@avnagovicihn ~]#
```

Рис. 4.1: Окно консоли

Устанавливаю git-flow. Для этого перехожу на роль супер-пользователя и прописываю команду(рис. 4.2).

```
[root@avnagovicihn ~]# dnf install gitflow
Copr repo for gitflow owned by elegos
Зависимости разрешены.
1.1 kB/s | 1.5 kB 00:01
-----
Пакет      Архитектура  Версия      Репозиторий      Размер
-----
Установка: gitflow      x86_64      1.12.3-1.fc34  copr:copr.fedorainfracloud.org:elegos:gitflow  57 k
-----
Результат транзакции
-----
Установка  1 Пакет
```

Рис. 4.2: Окно консоли

Перехожу к установке nodejs. Для этого прописываю команду (рис. 4.3).


```
[root@avnagovicihn ~]# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:02:16 назад, Пт 08 мар 2024 13:18:25.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия  Release      Размер
-----
Установка:
nodejs     x86_64       1:20.10.0-3.fc39  updates      48 k
Установка зависимостей:
nodejs-lib x86_64       1:20.10.0-3.fc39  updates      15 M
Установка слабых зависимостей:
nodejs-doc x86_64       1:20.10.0-3.fc39  updates      8.1 M
nodejs-full-libs x86_64 1:20.10.0-3.fc39  updates      8.5 M
nodejs-npm x86_64       1:10.2.3-1.20.10.0-3.fc39  updates      2.2 M
```

Рис. 4.3: Окно консоли

Проверяю корректность установки пакетов команды pnpm.(рис. 4.4).

```
[root@avnagovicihn ~]# dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:04:11 назад, Пт 08 мар 2024 13:18:25.
Пакет pnpm-8.12.0-1.fc39.poaarch уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 4.4: Окно консоли

Перехожу к настройке nodejs. Добавляю каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.(рис. 4.5).

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

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

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

Рис. 4.5: Окно консоли

4.2 Настройка общепринятых коммитов.

Устанавливаю программу, которая используется для помощи в форматировании коммитов (рис. 4.6).

```
[avnagovicihn@avnagovicihn ~]$ pnpm add -g commitizen

Update available! 8.12.0 → 8.15.4.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v8.15.4
Run "pnpm add -g pnpm" to update.

Follow @pnpmjs for updates: https://twitter.com/pnpmjs

Packages: +152
+++++
Downloading registry.npmjs.org/typescript/5.4.2: 5,82 MB/5,82 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

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

Done in 10.3s
```

Рис. 4.6: Окно консоли

Далее Устанавливаю программу, которая используется для помощи в создании логов (рис. 4.7).

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

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

Done in 5.2s
```

Рис. 4.7: Окно консоли

Создаю репозиторий на github (рис. 4.7).

```
[avnagovicihn@avnagovicihn ~]$ gh repo create git-extended --public
✓ Created repository AVNagov/git-extended on GitHub
https://github.com/AVNagov/git-extended
```

Рис. 4.8: Окно консоли

Копирую репозиторий себе на виртуальную машину, после чего перехожу в каталог.(рис. 4.9).

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

Рис. 4.9: Окно консоли

Создаю первый коммит и выкладываю на github (рис. 4.10).

```
[avnagovicihn@avnagovicihn git-extended]$ git commit -m "First commit"
Текущая ветка: main

Начальный коммит

нечего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
[avnagovicihn@avnagovicihn git-extended]$ git remote add origin git@github.com:AVNagov/git-extended.git
error: внешний репозиторий origin уже существует
[avnagovicihn@avnagovicihn git-extended]$ git push -u origin master
```

Рис. 4.10: Окно консоли

Перехожу к конфигурации общепринятых коммитов (рис. 4.11).

```
[avnagovicihn@avnagovicihn git-extended]$ pnpm init
Wrote to /home/avnagovicihn/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. 4.11: Окно консоли

Теперь перехожу к редактированию файла package.json (рис. 4.12).

```
[avnagovicihn@avnagovicihn git-extended]$ gedit package.json
```

Рис. 4.12: Окно консоли

Редактирую файл (рис. 4.13).

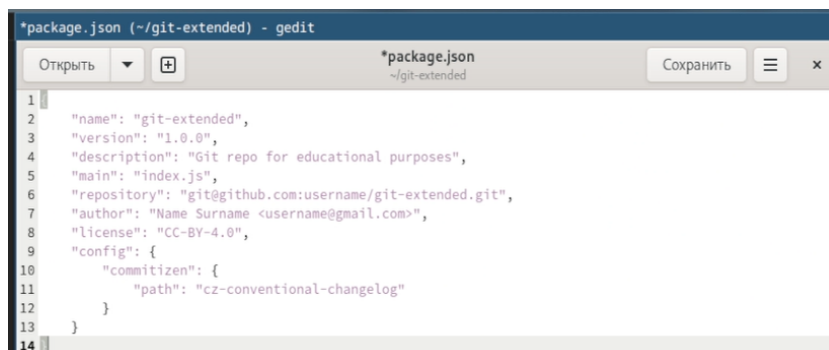


Рис. 4.13: Окно текстового редактора

Отправляю все github (рис. 4.14).

```
[avnagovicihn@avnagovicihn git-extended]$ git add .
[avnagovicihn@avnagovicihn git-extended]$ git cz
cz-cli@4.3.0, 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)
? Write a short, imperative tense description of the change (max 94 chars):
(2) 94
? 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 (корневой коммит) 7ebd833] feat: 94
1 file changed, 14 insertions(+)
create mode 100644 package.json
[avnagovicihn@avnagovicihn git-extended]$
[avnagovicihn@avnagovicihn git-extended]$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 432 байта | 432.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AVNagov/git-extended.git
 * [new branch]      main -> main
```

Рис. 4.14: Окно консоли

Инициализирую git-flow (рис. 4.15).

```

[avnagovicihn@avnagovicihn 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? []
Hooks and filters directory? [/home/avnagovicihn/git-extended/.git/hooks]
[avnagovicihn@avnagovicihn git-extended]$ git branch
* develop
  main
[avnagovicihn@avnagovicihn 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/AVNagov/git-extended/pull/new/develop
remote:
To github.com:AVNagov/git-extended.git
* [new branch]      develop -> develop

```

Рис. 4.15: Окно консоли

Проверяю, что нахожусь на ветке develop и загружаю весь репозиторий в хранилище (рис. 4.16).

```

[avnagovicihn@avnagovicihn git-extended]$ git branch
* develop
  main
[avnagovicihn@avnagovicihn 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/AVNagov/git-extended/pull/new/develop
remote:
To github.com:AVNagov/git-extended.git
* [new branch]      develop -> develop

```

Рис. 4.16: Окно консоли

Устанавливаю внешнюю ветку как вышестоящую для этой ветки и создаю релиз с версией 1.0.0 (рис. 4.17).

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

Рис. 4.17: Окно консоли

Создаю журнал изменений и добавляю журнал изменений в индекс (рис. 4.18).

```
[avnagovicihn@avnagovicihn git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[avnagovicihn@avnagovicihn git-extended]$ git add CHANGELOG.md
[avnagovicihn@avnagovicihn git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 6805b89] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 4.18: Окно консоли

Добавляю журнал изменений в индекс (рис. 4.19).

```
GNU nano 7.2 /home/avnagovicihn
Merge branch 'release/1.0.0'
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной вышестоящей
# ветки в тематическую ветку.
v 1.0.0
# Строки, начинающиеся с «#» будут проигнорированы, а пустое сообщение
# отменяет процесс коммита.
```

Рис. 4.19: Окно GNU nano

Добавляю журнал изменений в индекс (рис. 4.20).

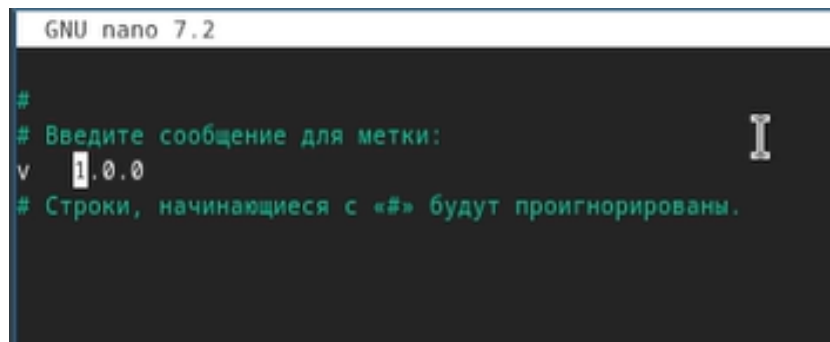


Рис. 4.20: Окно GNU nano

Добавляю журнал изменений в индекс (рис. 4.22).

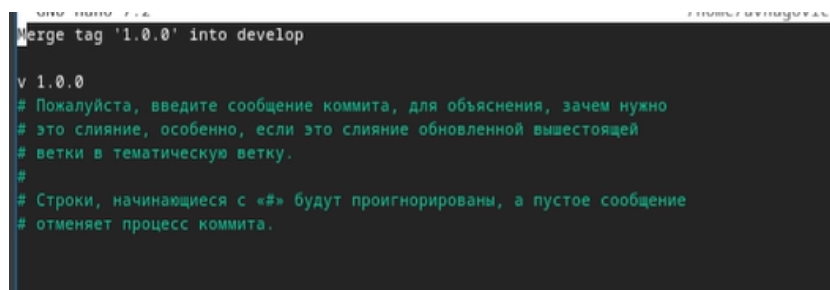


Рис. 4.21: Окно GNU nano

Загружаю релизную ветку в основную ветку и отправляю данные на github (рис. ??).

```

[avnagovicihn@avnagovicihn git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 637 байтов | 637.00 КиБ/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:AVNagov/git-extended.git
   7ebd833..944f28e  develop -> develop
   7ebd833..80b95f5  main -> main
[avnagovicihn@avnagovicihn git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 156 байтов | 156.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AVNagov/git-extended.git
   * [new tag]         1.0.0 -> 1.0.0
[avnagovicihn@avnagovicihn git-extended]$

```

Рис. 4.22: Окно консоли

Создаю релиз на github (рис. 4.23).

```

[avnagovicihn@avnagovicihn git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/AVNagov/git-extended/releases/tag/v1.0.0

```

Рис. 4.23: Окно консоли

Создаю ветку для новой функциональности и объединяю ветку feature_branch с develop (рис. 4.24).


```
[avnagovicihn@avnagovicihn 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

[avnagovicihn@avnagovicihn 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 944f28e).

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'
```

Рис. 4.24: Окно консоли

Создаю релиз с версией 1.2.3 (рис. 4.25).

```
[avnagovicihn@avnagovicihn 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. 4.25: Окно консоли

Обновляю номер версии в файле package.json (рис. 4.26).

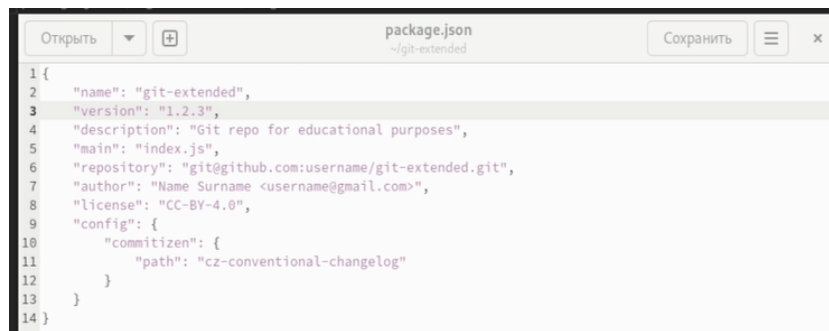


Рис. 4.26: Окно текстового редактора

Создаю журнал изменений и добавляю журнал изменений в индекс (рис. 4.27).

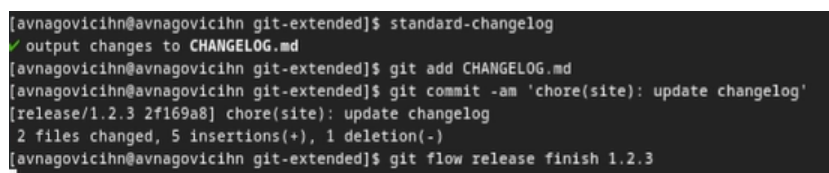


Рис. 4.27: Окно консоли

Добавляю журнал изменений в индекс (рис. 4.28).

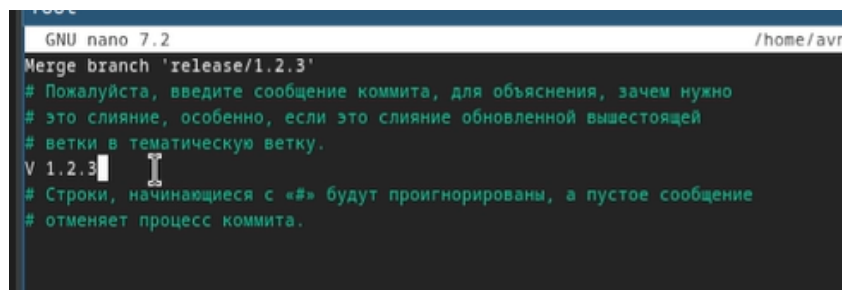


Рис. 4.28: Окно GNU nano

Добавляю журнал изменений в индекс (рис. 4.29).

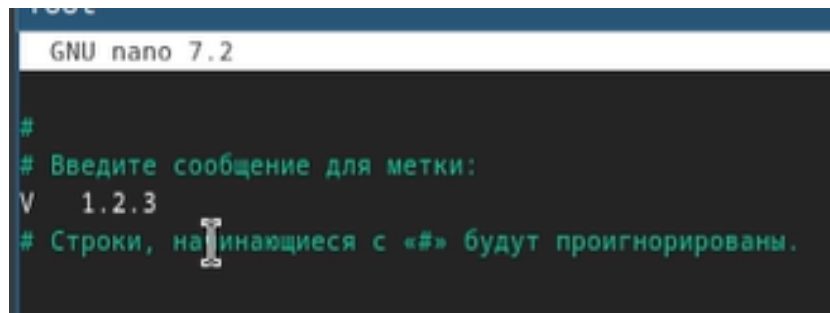


Рис. 4.29: Окно GNU nano

Добавляю журнал изменений в индекс (рис. 4.30).

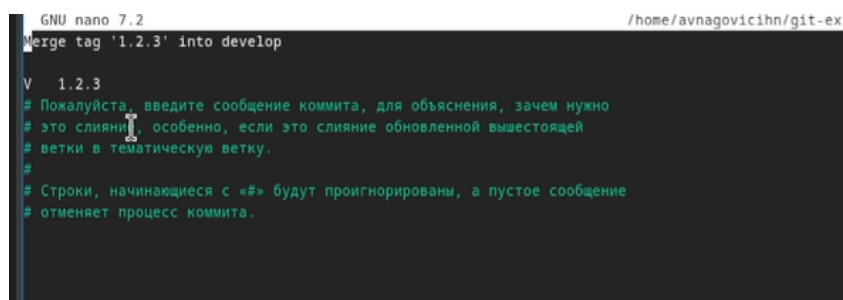


Рис. 4.30: Окно GNU nano

Загружаю релизную ветку в основную ветку и отправляю данные на github (рис. 4.31).

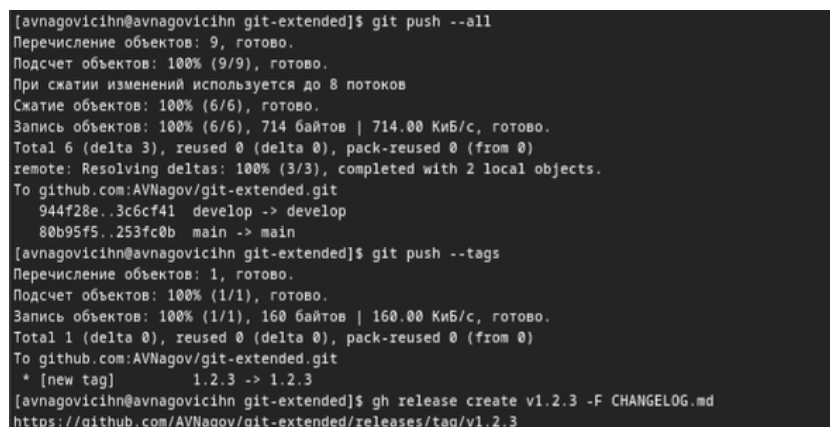


Рис. 4.31: Окно консоли

Проверяю правильность выполнения команды (рис. 4.32).

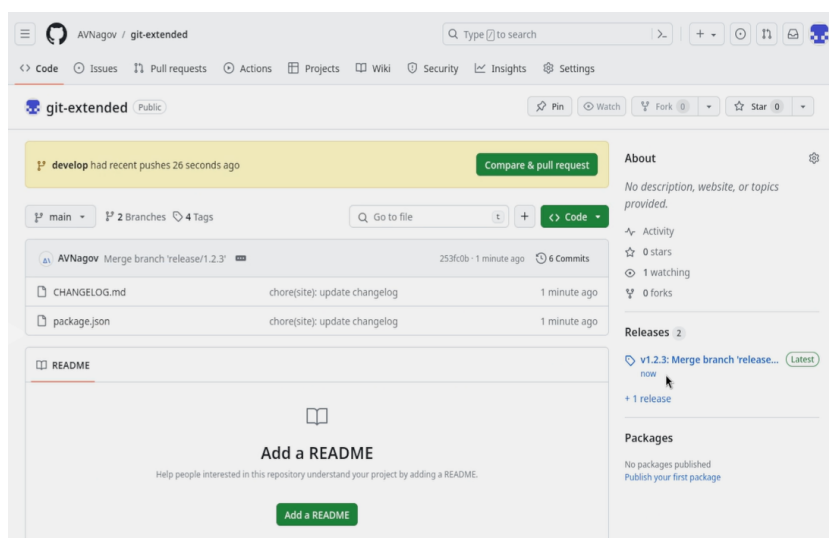


Рис. 4.32: Страница сайта github

5 Выводы

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