

Отчёт по лабораторной работе №4

Дисциплина: Операционные системы

Долгаев Евгений НММбд-01-24

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Практический сценарий использования git	9
4	Выводы	17
	Список литературы	18

Список таблиц

Список иллюстраций

3.1	Установка Gitflow	7
3.2	Установка Node.js	7
3.3	Настройка Node.js	8
3.4	Настройка Node.js	8
3.5	commitizen	9
3.6	standard-changelog	9
3.7	Репозиторий на Github	10
3.8	Первый коммит	10
3.9	Выполнение команды git push	10
3.10	Добавление новых файлов	11
3.11	Работа команд git cz и git push	11
3.12	Инициализация git-flow	11
3.13	Проверка	12
3.14	Загрузка репозитория	12
3.15	Установка внешней ветки	12
3.16	Создание релиза	12
3.17	Создание журнала изменений	12
3.18	Добавление журнала изменений в индекс	13
3.19	Релизная ветка сливается в основную	13
3.20	Отправка на Github	13
3.21	Отправка на Github	13
3.22	Новая ветка	14
3.23	Объединение веток	14
3.24	Создание релиза	15
3.25	Новая ветка	15
3.26	Журнал изменений	16
3.27	Добавление в индукс	16
3.28	Создание релиза	16

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 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Для начала установим Gitflow (рис. 3.1).

```
root@esdolghev:~# dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/ copr repo/elegos/gitflow/fedora-41/ 100% | 556.0 B/s | 652.0 B | 00m01s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является частью
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]: y
root@esdolghev:~# dnf install gitflow
Обновление и загрузка репозитория:
Fedora 41 openh264 (From Cisco) - x86_64 100% | 743.0 B/s | 989.0 B | 00m01s
Copr repo for gitflow owned by elegos 100% | 3.6 KiB/s | 2.4 KiB | 00m01s
Репозитории загружены.

Пакет Арх. Версия Репозиторий Размер
Установка:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedorainfracloud.org:elegos:gitflow 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]:
```

Рис. 3.1: Установка Gitflow

Далее установим Node.js и настроим его (рис. 3.2, 3.3, 3.4)

```
root@esdolghev:~# dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "nodejs-1:22.14.0-2.fc41.x86_64" уже установлен.

Нечего делать.
root@esdolghev:~# dnf install npm
Обновление и загрузка репозитория:
Репозитории загружены.

Пакет Арх. Версия Репозиторий Размер
Установка:
npm x86_64 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]:
```

Рис. 3.2: Установка Node.js

```
root@esdolgaev:~# pnpm setup
Appended new lines to /root/.bashrc

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

To start using pnpm, run:
source /root/.bashrc
root@esdolgaev:~#
```

Рис. 3.3: Настройка Node.js

```
source /root/.bashrc
root@esdolgaev:~# source ~/.bashrc
root@esdolgaev:~#
```

Рис. 3.4: Настройка Node.js

Настроим общепринятые коммиты (рис. 3.5, 3.6):

- 1) commitizen


```
root@esdolgaev:~# pnpm add -g commitizen

Update available! 9.13.0 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm self-update" to update.

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

Progress: resolved 137, reused 0, downloaded 124, added 0
```

Рис. 3.5: commitizen

2) standard-changelog

```
Done in 8.4s
root@esdolgaev:~# 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

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

Done in 5.2s
root@esdolgaev:~#
```

Рис. 3.6: standard-changelog

3.2 Практический сценарий использования git

Создим репозиторий на GitHub. Для примера назовём его git-extended (рис. 3.7).

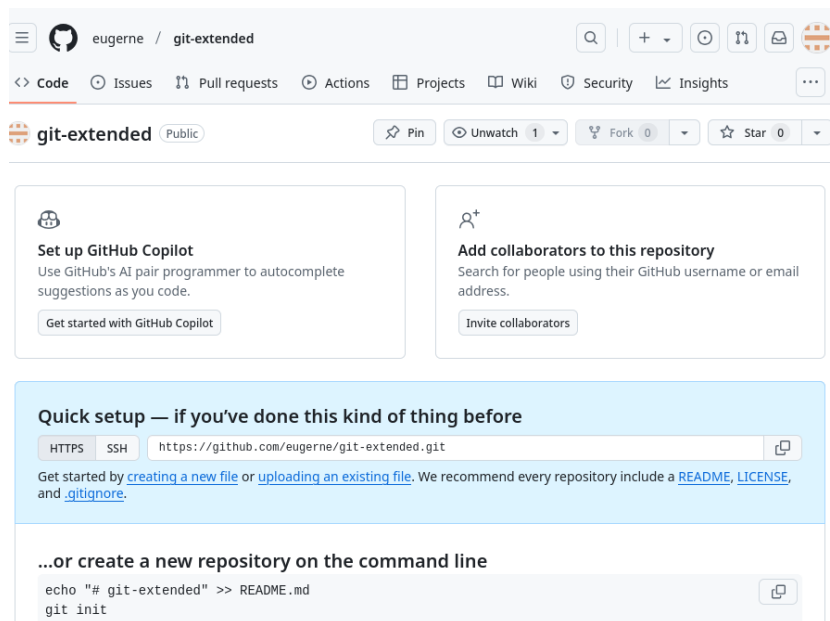


Рис. 3.7: Репозиторий на Github

Сделаем первый коммит (рис. 3.8, 3.9).

```
root@esdolgaev:~# git commit -m "first commit"
[master (корневой коммит) 57f1b46] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
root@esdolgaev:~#
```

Рис. 3.8: Первый коммит

```
root@esdolgaev:~# git push git@github.com:eugerne/git-extended.git master
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 880 байтов | 880.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:eugerne/git-extended.git
 * [new branch]      master -> master
root@esdolgaev:~# git push -u git@github.com:eugerne/git-extended.git master
branch 'master' set up to track 'git@github.com:eugerne/git-extended.git/master'.
Everything up-to-date
root@esdolgaev:~#
```

Рис. 3.9: Выполнение команды git push

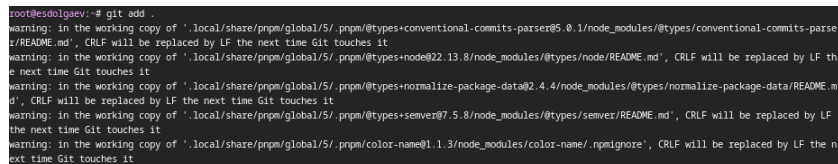
Далее произведем конфигурацию для пакетов Node.js с помощью команды:

```
pnpm init
```

Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов:

```
"config": {  
  "commitizen": {  
    "path": "cz-conventional-changelog"  
  }  
}
```

Добавим новые файлы (рис. 3.10):



```
root@esdolgayev:~# git add  
warning: in the working copy of '.local/share/pnpm/global/5/.pnpm/@types+conventional-commits-parser@5.0.1/node_modules/@types/conventional-commits-parser/README.md', CRLF will be replaced by LF the next time Git touches it  
warning: in the working copy of '.local/share/pnpm/global/5/.pnpm/@types+node@22.13.8/node_modules/@types/node/README.md', CRLF will be replaced by LF the next time Git touches it  
warning: in the working copy of '.local/share/pnpm/global/5/.pnpm/@types+normalize-package-data@2.4.4/node_modules/@types/normalize-package-data/README.md', CRLF will be replaced by LF the next time Git touches it  
warning: in the working copy of '.local/share/pnpm/global/5/.pnpm/@types+semver@7.5.8/node_modules/@types/semver/README.md', CRLF will be replaced by LF the next time Git touches it  
warning: in the working copy of '.local/share/pnpm/global/5/.pnpm/color-name@1.1.3/node_modules/color-name/README.md', CRLF will be replaced by LF the next time Git touches it
```

Рис. 3.10: Добавление новых файлов

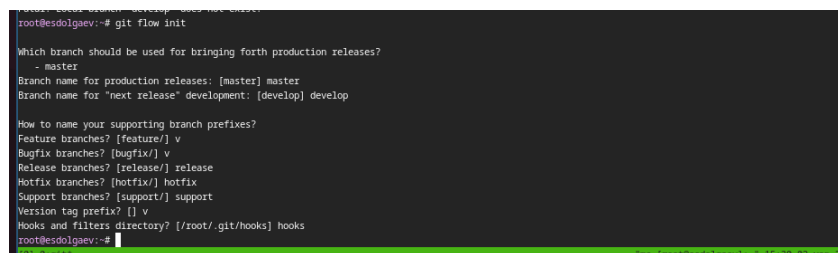
Выполним коммит и отправим на github (рис. 3.11):



```
root@esdolgayev:~# git cz  
Parsing JSON at /root/package.json for commitizen config failed:  
  
root@esdolgayev:~# git push  
Everything up-to-date  
root@esdolgayev:~#
```

Рис. 3.11: Работа команд git cz и git push

Инициализируем git-flow (рис. 3.12):



```
root@esdolgayev:~# git flow init  
Which branch should be used for bringing forth production releases?  
- master  
Branch name for production releases: [master] master  
Branch name for "next release" development: [develop] develop  
  
How to name your supporting branch prefixes?  
Feature branches? [feature/] v  
Bugfix branches? [bugfix/] v  
Release branches? [release/] release  
Hotfix branches? [hotfix/] hotfix  
Support branches? [support/] support  
Version tag prefix? [v] v  
Hooks and filters directory? [/root/.git/hooks] hooks  
root@esdolgayev:~#
```

Рис. 3.12: Инициализация git-flow

Проверим, что мы на ветке develop: (рис. 3.13):

```
root@esdolgaev:~# git branch
* develop
  master
root@esdolgaev:~#
```

Рис. 3.13: Проверка

Загрузим весь репозиторий в хранилище: (рис. 3.14):

```
root@esdolgaev:~# git push --all git@github.com:eugene/git-extended.git
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/eugene/git-extended/pull/new/develop
remote:
To github.com:eugene/git-extended.git
 * (new branch)      develop -> develop
root@esdolgaev:~#
```

Рис. 3.14: Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 3.15):

```
root@esdolgaev:~# git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
root@esdolgaev:~#
```

Рис. 3.15: Установка внешней ветки

Создадим релиз с версией 1.0.0 (рис. 3.16):

```
root@esdolgaev:~# git flow release start 1.0.0
Переключимся на новую ветку «release1.0.0»

Summary of actions:
- A new branch 'release1.0.0' was created, based on 'develop'
- You are now on branch 'release1.0.0'

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'

root@esdolgaev:~#
```

Рис. 3.16: Создание релиза

Создадим журнал изменений (рис. 3.17):

```
root@esdolgaev:~# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
root@esdolgaev:~#
```

Рис. 3.17: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 3.18):

```

root@esdolgaev:~# git add CHANGELOG.md
root@esdolgaev:~# git commit -m 'chore(site): add changelog'
[release1.0.0 d5491f6] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
root@esdolgaev:~#
0 0:git*

```

Рис. 3.18: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку (рис. 3.19):

```

root@esdolgaev:~# git flow release finish 1.0.0
Переключились на ветку «master»
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
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 release1.0.0 удалена (delta d5491f6).

Summary of actions:
- Release branch 'release1.0.0' has been merged into 'master'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release1.0.0' has been locally deleted
- You are now on branch 'develop'

```

Рис. 3.19: Релизная ветка сливается в основную

Отправим данные на github (рис. 3.20, 3.21):

```

root@esdolgaev:~# git push --all git@github.com:eugene/git-extended.git
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (5/5), готово.
При скатки коммитов используется до 2 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 2.62 Кб | 2.63 Мб/с, готово.
Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
To github.com:eugene/git-extended.git
 57f1b46..7e29d55 develop -> develop
 57f1b46..b798360 master -> master
root@esdolgaev:~#
0 0:git*

```

Рис. 3.20: Отправка на Github

```

root@esdolgaev:~# git push --tags git@github.com:eugene/git-extended.git
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 153 байта | 153.00 Кб/с, готово.
Total 1 (delta 0), reused 1 (delta 0), pack-reused 0 (from 0)
To github.com:eugene/git-extended.git
 * [new tag]
  v1.0.0 -> v1.0.0
root@esdolgaev:~#
0 0:git*

```

Рис. 3.21: Отправка на Github

Создадим ветку для новой функциональности (рис. 3.22):

```
root@esdolgaev:~# git flow feature start feature_branch
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
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

root@esdolgaev:~#
```

Рис. 3.22: Новая ветка

Далее, продолжаем работу с git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop (рис. 3.23):

```
root@esdolgaev:~# git flow feature finish feature_branch
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на ветку «develop»
Ваша ветка опережает «origin/develop» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Уже актуально.
Ветка vfeature_branch удалена (была 7e29d55).

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'

root@esdolgaev:~#
```

Рис. 3.23: Объединение веток

Создадим релиз с версией 1.2.3 (рис. 3.24):

```

root@esdolgaev:~# git flow release start 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на новую ветку «release1.2.3»

Summary of actions:
- A new branch 'release1.2.3' was created, based on 'develop'
- You are now on branch 'release1.2.3'

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. 3.24: Создание релиза

Обновим номер версии в файле package.json (рис. 3.25).

```

package.json  [-M--] 21 L:[ 1+ 2 3/ 16] *(51 / 386b) 0034
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:username/git-extended.git",
  "author": "Name Surname <username@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 3.25: Новая ветка

Создадим журнал изменений (рис. 3.26):

```
root@esdolgaev:~# standard-changelog
✓ output changes to CHANGELOG.md
root@esdolgaev:~#
```

Рис. 3.26: Журнал изменений

Добавим журнал изменений в индекс (рис. 3.27):

```
root@esdolgaev:~# git add CHANGELOG.md
git commit -am 'chore(site): update changelog'
[release1.2.3 effda35] chore(site): update changelog
1 file changed, 4 insertions(+)
root@esdolgaev:~#
```

Рис. 3.27: Добавление в индекс

Создадим релиз на github с комментарием из журнала изменений. Для этого будем использовать утилиты работы с github: (рис. 3.28):

```
root@esdolgaev:~# gh release create v1.2.3 -F CHANGELOG.md
https://github.com/eugene/git-extended/releases/tag/v1.2.3
root@esdolgaev:~#
```

Рис. 3.28: Создание релиза

4 Выводы

В ходе выполнения лабораторной работы я получил навыки управления репозиториями git.

Список литературы