

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера и операционные системы

Ванюшкина Татьяна Валерьевна

Содержание

1	Задание	5
2	Цель работы	6
3	Техническое обеспечение	7
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

Список таблиц

1 Задание

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

2 Цель работы

Целью данной работы является получение навыков правильной работы с репозиториями git.

3 Техническое обеспечение

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.

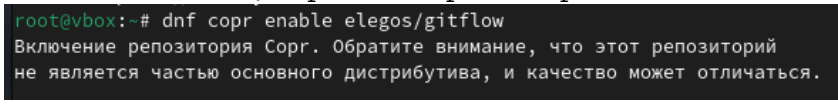
4 Выполнение лабораторной работы

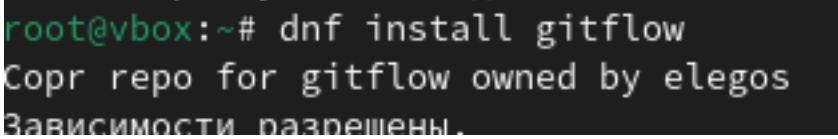
1. Установка программного обеспечения

Переключаемся на роль супер-пользователя

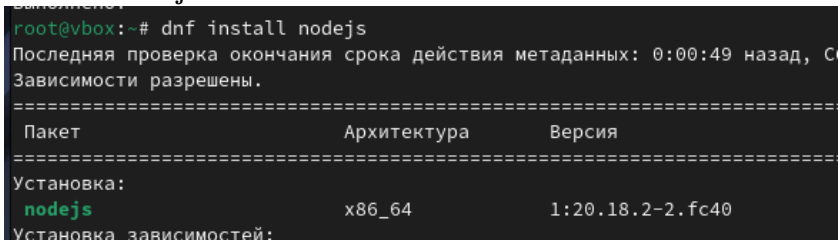
(рис.1 ??)  {#fig:002}

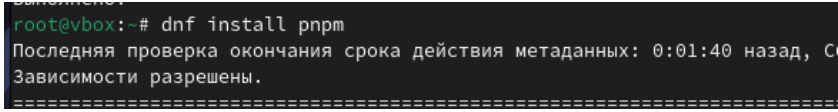
Устанавливаем из коллекции репозитория Copr

(рис.2 ??)  {#fig:003}

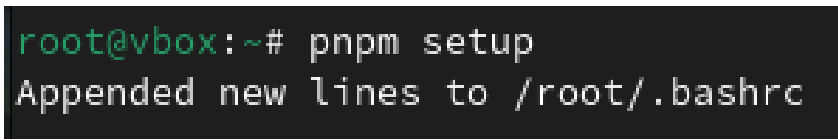
(рис.3 ??)  {#fig:004}

Устанавливаем Node.js

(рис.4 ??)  {#fig:005}

(рис.5 ??)  {#fig:006}

Запускаем :

(рис.6 ??)  {#fig:007}

Выполняем :

(рис.7 ??)  {#fig:008}

Общепринятые коммиты:

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

Update available! 9.0.6 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm add -g pnpm" to update.

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

(рис.8 ??)

{#fig:009}

```
root@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

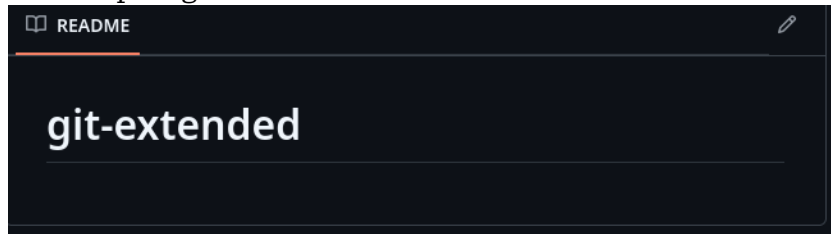
/root/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0
```

(рис.9 ??)

{#fig:010}

2.Практический сценарий использования git

Создаем репозиторий git-extended на GitHub



(рис.10 ??)

{#fig:011}

Клонируем его

```
root@vbox:~# git clone --recursive https://github.com/tatyana952/git-extended.git
Клонирование в «git-extended»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (3/3), готово.
```

(рис.11 ??)

{#fig:012}

Создание первого коммита



(рис.12 ??) {#fig:013}

(рис.13 ??) `root@vbox:~# git remote add origin https://github.com/tatyana952/git-extended.git` {#fig:015}

(рис.14 ??) `root@vbox:~/git-extended# git push -u origin master` {#fig:016}

Конфигурируем для пакетов Node.js

(рис.15 ??) `root@vbox:~/git-extended# pnpm init` {#fig:017}

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

```
package.json  [-----]  0 L:[ 1+ 0 1/ 18] *(0 / 461)
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:tatyana952/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Tatyana Vanyushkina <1132246713@pfur.ru>",
  "license": "CC-BY-4.0"
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

(рис.16 ??) {#fig:018}

Добавляем новые файлы и выполняем коммит:

```
root@vbox:~/git-extended# git add .
root@vbox:~/git-extended# git cz
```

(рис.17 ??) {#fig:019}

Отправляем файлы на гитхаб

```
root@vbox:~/git-extended# git push
Username for 'https://github.com': tatyana
```

(рис.18 ??) {#fig:020}

Инициализируем git-flow

```
root@vbox:~/git-extended# git flow init
```

(рис.19 ??) {#fig:021}

Проверяем, что мы на ветке develop:

```
root@vbox:~/git-extended# git branch
* main
```

(рис.20 ??) {#fig:022}

Загружаем весь репозиторий в хранилище

```
root@vbox:~/git-extended# git push --all
```

(рис.21 ??) {#fig:023}

Устанавливаем внешнюю ветку как вышестоящую для этой ветки:

```
root@vbox:~/git-extended# git branch --set-upstream-to=origin/develop develop
```

(рис.22 ??) {#fig:024}

Создадим релиз с версией 1.0.0

```
root@vbox:~/git-extended# git flow release start 1.0.0
```

(рис.23 ??) {#fig:025}

Создадим журнал изменений

```
root@vbox:~/git-extended# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

(рис.24 ??) {#fig:026}

Добавим журнал изменений в индекс

```
root@vbox:~/git-extended# git add CHANGELOG.md
root@vbox:~/git-extended# git commit -am 'chore(site): add changelog'
[main 8385525] chore(site): add changelog
2 files changed, 21 insertions(+)
create mode 100644 CHANGELOG.md
create mode 100644 package.json
```

(рис.25 ??) {#fig:027}

Зальём релизную ветку в основную ветку

```
root@vbox:~/git-extended# git flow release finish 1.0.0
```

(рис.26 ??) {#fig:028}

Отправим данные на github

```
root@vbox:~/git-extended# git push --all
```

(рис.27 ??) {#fig:029}

```
root@vbox:~/git-extended# git push --tags
```

(рис.28 ??) {#fig:030}

Создадим релиз на github. Для этого будем использовать утилиты работы с github:

```
root@vbox:~/git-extended# gh release create v1.0.0 -F CHANGELOG.md
```

(рис.29 ??) {#fig:031}

3. Работа с репозиторием git

Создадим ветку для новой функциональности

```
root@vbox:~# git flow feature start feature_branch
Переключились на новую ветку «feature/feature_branch»
```

(рис.30 ??) {#fig:032}

Объединяем ветку feature_branch с develop

```
root@vbox:~# git flow feature finish feature_branch
Переключились на ветку «develop»
Уже актуально.
```

(рис.31 ??) {#fig:033}

Создадим релиз с версией 1.2.3

```
root@vbox:~# git flow release start 1.2.3
Переключились на новую ветку «release/1.2.3»
```

(рис.32 ??) {#fig:034}

Создадим журнал изменений

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

(рис.33 ??) {#fig:035}

Добавим журнал изменений в индекс

```
root@vbox:~# git add CHANGELOG.md
root@vbox:~# git commit -am 'chore(site): update changelog'
[release/1.2.3 4731de2] chore(site): update changelog
1 file changed, 5 insertions(+)
create mode 100644 CHANGELOG.md
```

(рис.34 ??) {#fig:034}

Зальём релизную ветку в основную ветку и отправим данные на github

(рис.35 ??) root@vbox:~/git-extended# git push --all {#fig:037}

(рис.36 ??) root@vbox:~/git-extended# git push --tags {#fig:038}

5 Выводы

Я получила навыков правильной работы с репозиториями git.

Список литературы

Курс: Архитектура компьютеров и операционные системы. Раздел “Операционные системы” (02.03.00, УГСН) (rudn.ru)