

# **Лабораторная работа №2**

**Дисциплина: Операционные системы**

Савостин Олег

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
4.1	Установка программного обеспечения . . . . .	10
4.2	Практический сценарий использования git . . . . .	12
<b>5</b>	<b>Выводы</b>	<b>20</b>
	<b>Список литературы</b>	<b>21</b>

# Список иллюстраций

4.1	Установка git-flow . . . . .	10
4.2	Установка Node.js . . . . .	11
4.3	pnpm setup . . . . .	11
4.4	source ~/.bashrc . . . . .	12
4.5	Общепринятые коммиты . . . . .	12
4.6	Новый репозиторий . . . . .	13
4.7	Первый коммит, выкладываемый на гитхаб . . . . .	13
4.8	package.json . . . . .	14
4.9	Отправка на github . . . . .	14
4.10	git-flow . . . . .	15
4.11	git push -all . . . . .	15
4.12	Changelog . . . . .	16
4.13	Релиз на гитхабе . . . . .	16
4.14	Версия 1.2.3 . . . . .	17
4.15	package.json 1.2.3 . . . . .	17
4.16	Новый журнал изменений . . . . .	18
4.17	Журнал изменений . . . . .	18
4.18	Слияние новой ветки и релиз на гитхаб . . . . .	18
4.19	Слияние новой ветки и релиз на гитхаб . . . . .	19
4.20	Версии на гит . . . . .	19

## **Список таблиц**

# 1 Цель работы

Получить навыки правильной работы с репозиториями git.

## 2 Задание

1. Установка программного обеспечения
2. Практический сценарий использования git

## 3 Теоретическое введение

Основные ветки (master) и ветки разработки (develop)

Для фиксации истории проекта в рамках этого процесса вместо одной ветки master исп

При использовании библиотеки расширений git-flow нужно инициализировать структуру

```
git flow init
```

Для github параметр Version tag prefix следует установить в v.

После этого проверьте, на какой ветке Вы находитесь:

```
git branch
```

Функциональные ветки (feature)

Под каждую новую функцию должна быть отведена собственная ветка, которую можно отп

Как правило, ветки feature создаются на основе последней ветки develop.

Создание функциональной ветки

Создадим новую функциональную ветку:

```
git flow feature start feature_branch
```

Далее работаем как обычно.

Окончание работы с функциональной веткой

По завершении работы над функцией следует объединить ветку `feature_branch` с `develop`

```
git flow feature finish feature_branch
```

Ветки выпуска (release)

Когда в ветке `develop` оказывается достаточно функций для выпуска, из ветки `develop` создается ветка `release`. Благодаря тому, что для подготовки выпусков используется специальная ветка, одна команда позволяет создать ветку `release` и вернуть ее в `develop`.

Создать новую ветку `release` можно с помощью следующей команды:

```
git flow release start 1.0.0
```

Для завершения работы на ветке `release` используются следующие команды:

```
git flow release finish 1.0.0
```

Ветки исправления (hotfix)

Ветки поддержки или ветки `hotfix` используются для быстрого внесения исправлений в ветку `develop`. Наличие специальной ветки для исправления ошибок позволяет команде решать проблемы, не затрагивая ветку `develop`.

Ветку `hotfix` можно создать с помощью следующих команд:

```
git flow hotfix start hotfix_branch
```



По завершении работы ветка hotfix объединяется с master и develop:

```
git flow hotfix finish hotfix_branch
```

## 4 Выполнение лабораторной работы

### 4.1 Установка программного обеспечения

Сперва устанавливаю git-flow из коллекции репозитория Copr (рис. 4.1).

```
root@vbox:~# dnf install gitflow
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates          100% | 9.1 KiB/s | 23.3 KiB | 00m03s
Fedora 41 - x86_64 - Updates          100% | 149.1 KiB/s | 2.5 MiB | 00m17s
Copr repo for gitflow owned by elegos  100% | 519.0 B/s | 2.4 KiB | 00m05s
Репозитории загружены.
```

Рис. 4.1: Установка git-flow

Устанавливаю Node.js (рис. 4.2).

```

Ключ был успешно импортирован.
[1/3] Проверить файлы пакета
[2/3] Подготовить транзакцию
[3/3] Установка gitflow-0:1.12.3-1.fc34.x86_64
Завершено!
root@vbox:~# dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.

Пакет Арх. Версия Репозиторий Размер
Установка:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
Установка зависимостей:
nodejs-libs x86_64 1:22.14.0-2.fc41 updates 75.9 MiB
Установка слабых зависимостей:
nodejs-docs noarch 1:22.14.0-2.fc41 updates 93.7 MiB
nodejs-full-libs x86_64 1:22.14.0-2.fc41 updates 30.4 MiB
nodejs-rpm x86_64 1:10.9.2-1.22.14.0-2.fc41 updates 9.3 MiB

Сводка транзакции:
Установка: 5 пакетов

Общий размер входящих пакетов составляет 39 MiB. Необходимо загрузить 39 MiB.
После этой операции будут использоваться дополнительные 210 MiB (установка 210 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/5] nodejs-1:22.14.0-2.fc41.x86_64
[2/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[3/5] nodejs-docs-1:22.14.0-2.fc41.noarch
[4/5] nodejs-rpm-1:10.9.2-1.22.14.0-2.fc41.x86_64
-----
[1/5] Total
-----
[1/5] nodejs-1:22.14.0-2.fc41.x86_64
[2/5] nodejs-libs-1:10.9.2-1.22.14.0-2.fc41.x86_64
[3/5] nodejs-docs-1:22.14.0-2.fc41.noarch
[4/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/5] nodejs-full-libs-1:22.14.0-2.fc41.x86_64
[5/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[5/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
[5/5] nodejs-libs-1:22.14.0-2.fc41.x86_64
-----
[5/5] Total
-----
Выполнение транзакции
[1/7] Проверить файлы пакета
[2/7] Подготовить транзакцию
[3/7] Установка nodejs-libs-1:22.14.0-2.fc41.x86_64
[3/7] Установка nodejs-libs-1:22.14.0-2.fc41.x86_64
[4/7] Установка nodejs-1:22.14.0-2.fc41.x86_64
[5/7] Установка nodejs-rpm-1:10.9.2-1.22.14.0-2.fc41.x86_64
[6/7] Установка nodejs-full-libs-1:22.14.0-2.fc41.x86_64
[7/7] Установка nodejs-docs-1:22.14.0-2.fc41.noarch
Завершено!

```

Рис. 4.2: Установка Node.js

После установки начинаю настраивать программы. Настраиваю Node.js ввожу команду `npm setup` и добавляю каталог с исполняемыми файлами (рис. 4.3).

```

root@vbox:~# npm setup
No changes to the environment were made. Everything is already up to date.

```

Рис. 4.3: npm setup

Перелогинюсь. (рис. 4.4).

```
root@vbox:~# source ~/.bashrc
root@vbox:~# pnpm setup
No changes to the environment were made. Everything is already up to date.
root@vbox:~# 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

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
Progress: resolved 151, reused 0, downloaded 150, added 150
```

Рис. 4.4: source ~/.bashrc

Используя общепринятые коммиты (рис. 4.5).

```
root@vbox:~# source ~/.bashrc
root@vbox:~# pnpm setup
No changes to the environment were made. Everything is already up to date.
root@vbox:~# 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

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
Progress: resolved 151, reused 0, downloaded 150, added 150
```

Рис. 4.5: Общепринятые коммиты

## 4.2 Практический сценарий использования git

Начинаю подключение репозитория к github. Создаю репозиторий на GitHub. Называю его git-extended (рис. 4.6).

```

root@vbox:~# mkdir git-extended
root@vbox:~# mc git-extended

root@vbox:~# cd git-extended
root@vbox:~/git-extended# gh repo create
? What would you like to do? Create a new repository on GitHub from scratch
? Repository name git-extended
? Description LabWork
? Visibility Public
? Would you like to add a README file? No
? Would you like to add a .gitignore? No
? Would you like to add a license? No
? This will create "git-extended" as a public repository on GitHub. Continue? Yes
✓ Created repository osavostin/git-extended on GitHub
https://github.com/osavostin/git-extended
? Clone the new repository locally? Yes
root@vbox:~/git-extended# mc

root@vbox:~/git-extended#

```

Рис. 4.6: Новый репозиторий

Делаю первый коммит (рис. 4.7).

```

root@vbox:~/git-extended/git-extended# git push -u origin master
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 367 байтов | 367.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:osavostin/git-extended.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

root@vbox:~/git-extended/git-extended# git commit -m "first commit"
Текущая ветка: master

Начальный КОММИТ

нечего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
root@vbox:~/git-extended/git-extended#

```

Рис. 4.7: Первый коммит, выкладываемый на гитхаб

Конфигурирую пакеты Node.js и заполняю несколько параметров пакета. Добавляю в файл package.json команду для формирования коммитов/. Вид package.json (рис. 4.8).

```
package.json  [-M--]  3 L:[ 1+ 6 7/ 15] *(190 / 348b) 0097 0x0
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:osavostin/git-extended.git",
  "author": "Oleg Savostin mr.sans.san2@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.8: package.json

Добавляю новые файлы, выполняю коммит cz и отправляю на github (рис. 4.9).

```
root@vbox:~/git-extended/git-extended# git add .
root@vbox:~/git-extended/git-extended# git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: docs: Documentation only changes
? 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):
(4) test
? 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
[master 0e7b2cf] docs: test
1 file changed, 9 insertions(+), 7 deletions(-)
root@vbox:~/git-extended/git-extended# git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 464 байта | 33.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:osavostin/git-extended.git
 b23e519..0e7b2cf master -> master
root@vbox:~/git-extended/git-extended# git flow init
```

Рис. 4.9: Отправка на github

Теперь инициализирую git-flow, префикс для ярлыков устанавливаю в v (рис. 4.10).

```

root@vbox:~# cd git-extended/git-extended
root@vbox:~/git-extended/git-extended# 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/] bug
Release branches? [release/] release
Hotfix branches? [hotfix/] hot
Support branches? [support/] support
Version tag prefix? [] v
Hooks and filters directory? [/root/git-extended/git-extended/.git/hooks]
root@vbox:~/git-extended/git-extended# cd
root@vbox:~# cd git-extended/git-extended
root@vbox:~/git-extended/git-extended# ls
package.json
root@vbox:~/git-extended/git-extended#

```

Рис. 4.10: git-flow

Проверяю, если я на ветке develop и загружаю весь репозиторий в хранилище. Устанавливаю внешнюю ветку как вышестоящую для этой ветки. Создаю релиз с версией 1.0.0 и создаю журнал изменений (рис. 4.11).

```

* develop
master
root@vbox:~/git-extended/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/osavostin/git-extended/pull/new/develop
remote:
To github.com:osavostin/git-extended.git
 * [new branch]   develop -> develop
root@vbox:~/git-extended/git-extended# git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
root@vbox:~/git-extended/git-extended# git flow release start 1.0.0
Переключились 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@vbox:~/git-extended/git-extended# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
root@vbox:~/git-extended/git-extended#

```

Рис. 4.11: git push --all

Добавляю журнал изменений в индекс и заливаю релизную ветку в основную

ветку (рис. 4.12).

```
root@vbox:~/git-extended/git-extended# git add CHANGELOG.md
root@vbox:~/git-extended/git-extended# git commit -am 'chore(site): add changelog'
[release1.0.0 dd25662] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 4.12: Changelog

Отправляю все данные на гитхаб и создаю релиз на гитхаб (рис. 4.13).

```
[root@vbox git-extended]# git flow release finish 1.0.0
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/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
To github.com:osavostin/git-extended.git
- [deleted]          release1.0.0
Ветка release1.0.0 удалена (была dd25662).

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; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
[root@vbox git-extended]#
```

Рис. 4.13: Релиз на гитхабе

Приступаем ко второму этапу Разработка новой функциональности. Создаю ветку для новой функциональности. Объединяю ветку feature\_branch с develop. Создаю релиз с версией 1.2.3 (рис. 4.14).



```
[root@vbox git-extended]# git flow feature finish feature_branch
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Уже на «develop»
Ваша ветка опережает «origin/develop» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Merge made by the 'ort' strategy.
Ветка vfeature_branch удалена (была ef7fce8).

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@vbox git-extended]# 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'

[root@vbox git-extended]#
```

Рис. 4.14: Версия 1.2.3

Обновляю номер версии в файле package.json и устанавливаю её в 1.2.3 (рис. 4.15).

```
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:osavostin/git-extended.git",
  "author": "Oleg Savostin mr.sans.san2@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.15: package.json 1.2.3

Создаю журнал изменений и добавляю его в индекс(рис. 4.16) (рис. 4.17)

```
[root@vbox git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@vbox git-extended]# git add CHANGELOG.md
[root@vbox git-extended]# git com
```

Рис. 4.16: Новый журнал изменений

```
[root@vbox git-extended]# gh release create v1.2.3 -F CHANGELOG.md
https://github.com/osavostin/git-extended/releases/tag/v1.2.3
[root@vbox git-extended]#
```

Рис. 4.17: Журнал изменений

Заливаю ветку в основную ветку и отправляю данные на гитхаб, после чего создаю релиз на гитхаб с комментарием из журнала изменений (рис. 4.18) (рис. 4.19) (рис. 4.20).

```
branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на ветку «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
/же на «master»
Ваша ветка опережает «origin/master» на 7 коммитов.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Ваша ветка опережает «origin/develop» на 5 коммитов.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release1.2.3 удалена (была 8d6a97c).

Summary of actions:
- Release branch 'release1.2.3' has been merged into 'master'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release1.2.3' has been locally deleted
- You are now on branch 'develop'
[root@vbox git-extended]#
```

Рис. 4.18: Слияние новой ветки и релиз на гитхаб

```
[root@vbox git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@vbox git-extended]# git add CHANGELOG.md
[root@vbox git-extended]# git commit -am 'chore(site):update changelog'
[release1.2.3 8d6a97c] chore(site):update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 4.19: Слияние новой ветки и релиз на гитхаб

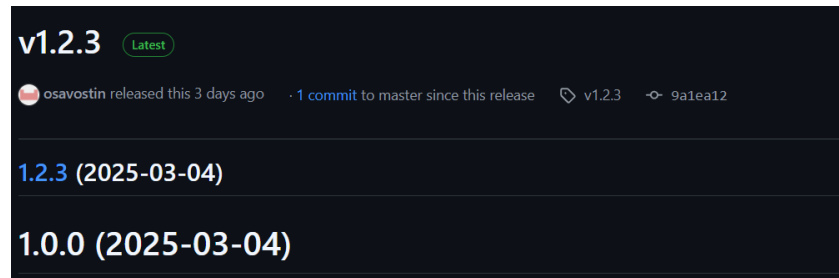


Рис. 4.20: Версии на гит

## 5 Выводы

Я получил навыки правильной работы с репозиториями git.

# **Список литературы**

Лабораторная работа номер 4