

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: *Операционные системы*

Студент: Николаев Дмитрий Иванович

Группа: НКНбд-01-20

МОСКВА

2021 г.

Цель работы:

Изучить идеологию и применение средств контроля версий. Научиться использовать систему контроля версий в своих проектах.

Ход работы:

Настроил систему git на своём компьютере.

Создание структуры каталога (M2)

```
[dinikolaev@dinikolaev ~]$ cd
[dinikolaev@dinikolaev ~]$ mkdir work
[dinikolaev@dinikolaev ~]$ ls
work  Документы  Изображения  Общедоступные  Шаблоны
Видео  Загрузки  Музыка  'Рабочий стол'
[dinikolaev@dinikolaev ~]$ cd work
[dinikolaev@dinikolaev work]$ mkdir 2021
[dinikolaev@dinikolaev work]$ ls
2021
[dinikolaev@dinikolaev work]$ cd 2021
[dinikolaev@dinikolaev 2021]$ mkdir 'Операционные системы'
[dinikolaev@dinikolaev 2021]$ ls
'Операционные системы'
[dinikolaev@dinikolaev 2021]$ cd 'Операционные системы'
[dinikolaev@dinikolaev Операционные системы]$ mkdir laboratory
[dinikolaev@dinikolaev Операционные системы]$ ls
laboratory
[dinikolaev@dinikolaev Операционные системы]$ cd laboratory
[dinikolaev@dinikolaev laboratory]$
```

Создал пустой репозиторий на github и заготовку README, после сделал первый коммит

```
[dinikolaev@dinikolaev laboratory]$ git init
Инициализирован пустой репозиторий Git в /home/dinikolaev/work/2021/Операционные
системы/laboratory/.git/
[dinikolaev@dinikolaev laboratory]$ echo "# Лабораторные работы" >> README.md
[dinikolaev@dinikolaev laboratory]$ git add README.md
[dinikolaev@dinikolaev laboratory]$ git commit -m "first commit"
[master (корневой коммит) ba678e6] first commit
Committer: dinikolaev <dinikolaev@dinikolaev.localdomain>
Ваше имя или электронная почта настроены автоматически на основании вашего
имени пользователя и имени машины. Пожалуйста, проверьте, что они
определены правильно.
Вы можете отключить это уведомление установив их напрямую. Запустите следующую
команду и следуйте инструкциям вашего текстового редактора, для
редактирования вашего файла конфигурации:

    git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 README.md
[dinikolaev@dinikolaev laboratory]$
```

Выложил первый коммит на github

```
[dinikolaev@dinikolaev laboratory]$ git remote add origin https://github.com/MrShogun/os-intro.git
[dinikolaev@dinikolaev laboratory]$ git push -u origin master
Gtk-Message: 20:49:33.473: Failed to load module "canberra-gtk-module"
Gtk-Message: 20:53:42.296: Failed to load module "canberra-gtk-module"
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/MrShogun/os-intro.git/'
[dinikolaev@dinikolaev laboratory]$ git push -u origin master
Gtk-Message: 20:54:33.165: Failed to load module "canberra-gtk-module"
Gtk-Message: 20:55:41.882: Failed to load module "canberra-gtk-module"
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/MrShogun/os-intro.git/'
[dinikolaev@dinikolaev laboratory]$ git push -u origin master
Gtk-Message: 20:56:22.861: Failed to load module "canberra-gtk-module"
Gtk-Message: 20:56:29.963: Failed to load module "canberra-gtk-module"
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 245 bytes | 245.80 KiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MrShogun/os-intro.git
 * [new branch]      master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
[dinikolaev@dinikolaev laboratory]$
```

Добавил файл лицензии и шаблон игнорируемых файлов

```
[dinikolaev@dinikolaev laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt
--2021-05-07 21:00:09-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)_ 104.20.150.16, 172.67.34.140, 104.20.151.16, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.150.16]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «legalcode.txt»

legalcode.txt          [ <=> ] 18,22K --.-KB/s  за ...
2021-05-07 21:00:10 (15,9 MB/s) - «legalcode.txt» сохранён [18657]
[dinikolaev@dinikolaev laboratory]$ curl -L -s https://www.gitignore.io
```

Добавил шаблон C++, добавил новые файлы, сделал второй коммит

```
[dinikolaev@dinikolaev laboratory]$ curl -L -s https://www.gitignore/api/c++ >>
.gitignore
[dinikolaev@dinikolaev laboratory]$ git add .
[dinikolaev@dinikolaev laboratory]$ git commit -am "second commit"
[master 987b657] second commit
  Committer: dinikolaev <dinikolaev@dinikolaev.localdomain>
  Ваше имя или электронная почта настроены автоматически на основании вашего
  имени пользователя и имени машины. Пожалуйста, проверьте, что они
  определены правильно.
  Вы можете отключить это уведомление установив их напрямую. Запустите следующую
  команду и следуйте инструкциям вашего текстового редактора, для
  редактирования вашего файла конфигурации:

    git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью к

    git commit --amend --reset-author

3 files changed, 396 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .gitignore.
 create mode 100644 legalcode.txt
[dinikolaev@dinikolaev laboratory]$
```

Отправил на github

```
[dinikolaev@dinikolaev laboratory]$ git push
Gtk-Message: 21:10:06.382: Failed to load module "canberra-gtk-module"
Gtk-Message: 21:10:21.562: Failed to load module "canberra-gtk-module"
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 6.09 KiB | 3.05 MiB/s, готово.
Total 4 (delta 8), reused 0 (delta 0), pack-reused 8
To https://github.com/MrShogun/os-intro.git
   ba670e6..987b657  master -> master
```

Инициализировал git-flow (настроил конфигурацию)

```
'gitflow/git-flow-release' -> '/usr/local/bin/git-flow-release'
'gitflow/git-flow-support' -> '/usr/local/bin/git-flow-support'
'gitflow/git-flow-version' -> '/usr/local/bin/git-flow-version'
'gitflow/gitflow-common' -> '/usr/local/bin/gitflow-common'
'gitflow/gitflow-shFlags' -> '/usr/local/bin/gitflow-shFlags'
[dinikolaev@dinikolaev laboratory]$ 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/] ^Z
[1]+  Остановлен  git flow init
[dinikolaev@dinikolaev laboratory]$ git flow init

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
```

Проверил, что я на ветке develop и создал релиз версии 1.0.0

```
[dinikolaev@dinikolaev laboratory]$ git branch
* develop
  master
[dinikolaev@dinikolaev laboratory]$ git flow release start 1.0.0
Переключено 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'
```

Записал версию и добавил файлы (на эту ветку)

```
[dinikolaev@dinikolaev laboratory]$ echo "1.0.0" >> VERSION
[dinikolaev@dinikolaev laboratory]$ git add .
warning: добавление встроенного git репозитория: gitflow
подсказка: You've added another git repository inside your current repository.
подсказка: Clones of the outer repository will not contain the contents of
подсказка: the embedded repository and will not know how to obtain it.
подсказка: If you meant to add a submodule, use:
подсказка:
подсказка:     git submodule add <url> gitflow
подсказка:
подсказка: If you added this path by mistake, you can remove it from
подсказка: index with:
подсказка:
подсказка:     git rm --cached gitflow
подсказка:
подсказка: See "git help submodule" for more information.
```

Добавил в индекс и залил релизную ветку в основную

```
[dinikolaev@dinikolaev laboratory]$ git commit -am 'third commit: add version'
[release/1.0.0 fa823aa] third commit: add version
Committer: dinikolaev <dinikolaev@dinikolaev.localdomain>
Ваше имя или электронная почта настроены автоматически на основании вашего
имени пользователя и имени машины. Пожалуйста, проверьте, что они
определены правильно.
Вы можете отключить это уведомление установив их напрямую. Запустите следующую
команду и следуйте инструкциям вашего текстового редактора, для
редактирования вашего файла конфигурации:

    git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью команды:

    git commit --amend --reset-author

3 files changed, 80 insertions(+)
create mode 100644 VERSION
create mode 100888 gitflow
create mode 100755 gitflow-installer.sh
[dinikolaev@dinikolaev laboratory]$ git flow release finish 1.0.0
```


Запустил все ветки на github

```
[dinikolaev@dinikolaev laboratory]$ git push --all
Gtk-Messsage: 88:44:19.128: Failed to load module "canberra-gtk-module"
Gtk-Messsage: 88:44:19.995: Failed to load module "canberra-gtk-module"
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (15/15), готово.
Сжатие объектов: 100% (15/15), готово.
Запись объектов: 100% (15/15), 7.68 KiB | 2.56 MiB/s, готово.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/MrShogun/os-intro.git
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
 * [new branch]      master -> master (fetch first)

подсказка: Обновления были отклонены, так как внешний репозиторий
подсказка: изменения, которых у вас нет в вашем локальном репозитории
подсказка: Обычно, это связано с тем, что кто-то уже отправил и
подсказка: то же место. Перед повторной отправкой ваших изменений
подсказка: забрать и слить изменения из внешнего репозитория себе
подсказка: (например, с помощью «git pull --»).
подсказка: Для дополнительной информации, смотрите «Note about f
подсказка: а «git push --help».
```

```
[dinikolaev@dinikolaev laboratory]$ git push --tags
Gtk-Messsage: 88:44:37.481: Failed to load module "canberra-gtk-a
Gtk-Messsage: 88:44:47.738: Failed to load module "canberra-gtk-a
Everything up-to-date
[dinikolaev@dinikolaev laboratory]$
```

Контрольные вопросы:

- 1) Системы контроля версий (Version Control System, VCS) - это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - с проектами. Чаще всего применяются при работе нескольких человек над одним проектом.
- 2) Хранилище в системе контроля версий - это удаленный репозиторий, где хранятся все файлы проекта.
Commit - это фиксация изменений перед загрузкой файлов в систему контроля версий.
История хранит все изменения проекта, и в случае необходимости позволяет откатиться к нужному месту.
Рабочая копия - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, необходимо загрузить новую версию проекта себе на компьютер.
- 3) Централизованные системы контроля версий хранят данные о проекте на едином сервере, и в случае его отключения, доступ к данным будет утерян (Perforce). В децентрализованных системах у каждого из участников проекта на компьютере хранится полная копия проекта. Среди централизованных VCS наиболее известны CVS, Subversion, а среди децентрализованных — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.
- 4) Сначала надо создать и подключить удаленный репозиторий. Затем, так как никто кроме пользователя не изменяет проект, то по мере изменения проекта пушить изменения на сервер, из-за чего нет необходимости загружать изменения.

- 5) Каждый раз перед работой с проектом нужно загрузить актуальную версию проекта на свой компьютер, и только потом работать над ним. После работы необходимо закоммитить изменения и запустить на сервер.
- 6) Основные задачи: упрощение обмена информацией, ускорение разработки, устранение ошибок и недоработок во время разработки (защита от случайных исправлений и удалений).
- 7)
 1. git init - инициализирует локальный репозиторий
 2. git add - добавляет файлы в репозиторий
 3. git commit - коммит версии (сохранение изменений)
 4. git pull - загружает актуальную версию проекта
 5. git push - отправляет измененный проект на сервер
 6. git checkout - позволяет переключаться между ветками
 7. git status - текущий статус проекта
 8. git branch - просмотр доступных веток
 9. git remote add - добавление удаленного репозитория
 10. git diff - просмотр текущих изменений
 11. git rm - удаление файла и/или каталог из индекса репозитория.
- 8) git branch,(→) git init,(→) git remote add,(→) git add,(→) git commit,(→) git push (git push --all).
- 9) Ветки нужны для разделения разработки. Например, можно разработать часть проекта в отдельной ветке, а не в основном проекте. В случае успешной разработки, эту ветку сливают с основной. Так убираются риски багов, ошибок, а также утечки данных.
- 10) Есть временные и системные файлы, которые засоряют проект и не нужны. Игнорируемые файлы – это, как правило, специфичные для платформы файлы или автоматически созданные файлы из систем сборки. Путь к ним можно добавить в файл .gitignore, тогда они не будут добавляться в проект.

Вывод:

Выполняя данную лабораторную работу, я изучил идеологию и применение средств контроля версий, создал аккаунт и репозиторий на github, выполнил важнейшие команды: инициализирование локального репозитория, подключение удаленных репозиторий, добавление и удаление нужных файлов, синхронизирование данных. Также я научился использовать Git Flow, который очень сильно упрощает разработку проектов и навигацию между ветками.