Цель работы:

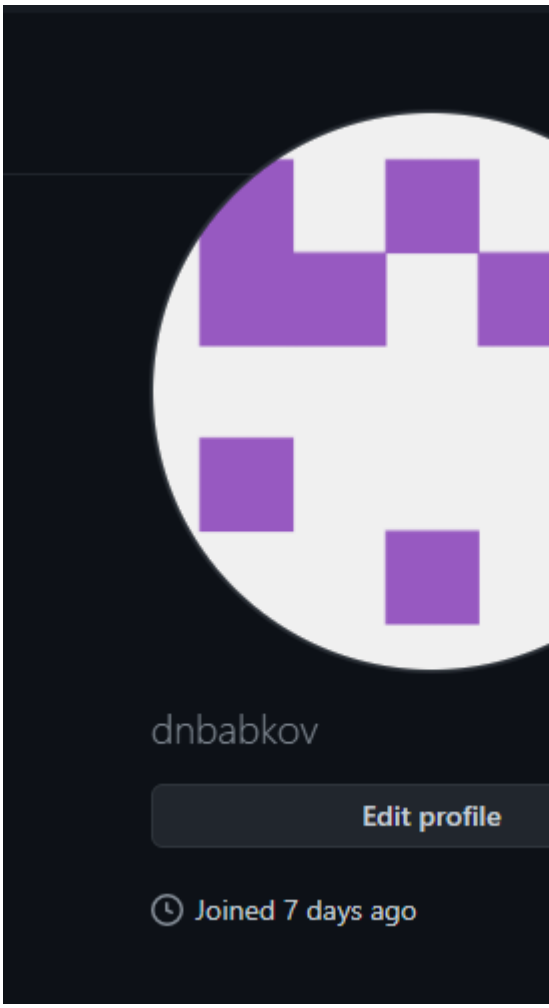
Изучить идеологию и применение средств контроля версий.

Задача:

Настроить репозиторий на GitHub и научиться работать с ним.

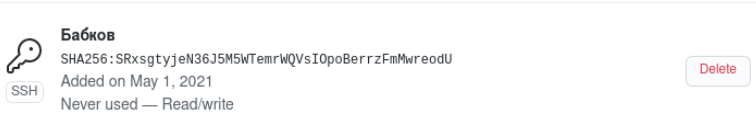
Выполнение лабораторной работы:

Настройка git.



1. Создадим учетную запись на https://github.org (Рис.1):
2. Настраиваем систему контроля версий git:
    1. С помощью команды ssh-keygen -C “Бабков Дмитрий 1032201726@pfur.ru” генерируем ssh ключ (Рис.2)

2. Добавляем этот ключ в свойствах github (Рис.3)

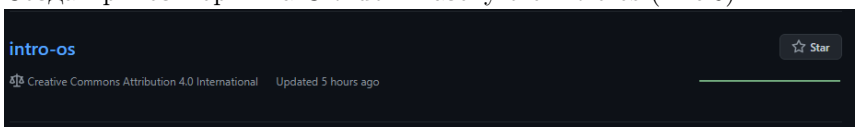


3. При помощи команды mkdir -p /home/work/2020-2021/OperatingSystems/laboratory создаем структуру каталога для лабораторных работ (Рис.4).

```
[dnbabkov@dnbabkov ~]$ mkdir -p ~/work/2020-2021/OperatingSystems/laboratory
[dnbabkov@dnbabkov ~]$ cd laboratory
```

Подключение репозитория к Github

- Создам рипозиторий на Github и назову его intro-os (Рис.5)



- С помощью команды git init инициализирую системы git (Рис.6)

```
[dnbabkov@dnbabkov laboratory]$ git init
Initialized empty Git repository in /home/dnbabkov/work/2020-2021/OperatingSystems/laboratory/.git/
```

- Создаем заготовку для файла README.md с помощью команд echo "# Лабораторные работы" >> README.md и git add README.md (Рис.7)

```
[dnbabkov@dnbabkov laboratory]$ echo "# Лабораторные работы" >> README.md
[dnbabkov@dnbabkov laboratory]$
[dnbabkov@dnbabkov laboratory]$ git add README.md
```

- Делаем первый коммит и загружаем его на Github (Рис.8,9)

```
[dnbabkov@dnbabkov laboratory]$ git commit -m "first commit"
[master (root-commit) 12f87e5] first commit
 Committer: Dmitry Nikolaevich Babkov <dnbabkov@dnbabkov.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 README.md
[dnbabkov@dnbabkov laboratory]$ git remote add origin git@github.com:dnbabkov/intro-os.git
[dnbabkov@dnbabkov laboratory]$ git push -u origin master
Warning: Permanently added the RSA host key for IP address '140.82.121.4' to the list of known hosts.
Counting objects: 3, done.
Writing objects: 100% (3/3), 260 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:dnbabkov/intro-os.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Первичная конфигурация

- Добавим файл лицензии (Рис.10)

```
[dnbabkov@dnbabkov laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-05-01 16:11:16--  https://creativecommons.org/licenses/by/4.0/legalcode.txt
Resolving creativecommons.org (creativecommons.org)... 104.20.150.16, 104.20.151.16, 172.67.34.140, ...
Connecting to creativecommons.org (creativecommons.org)|104.20.150.16|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: 'LICENSE'

    [ <=>                   ] 18,657      --.-K/s   in 0s

2021-05-01 16:11:16 (190 MB/s) - 'LICENSE' saved [18657]
```

- Также с помощью команды Curl -L -s https://www.gitignore.io/api/c >> .gitignore добавляем шаблон игнорируемых файлов для C

- С помощью git add . добавляем новые файлы (Рис11), делаем коммит командой git commit -a (Рис.12, 13) и отправляем на Гитхаб командой git push (Рис. 14)

```
[dnbabkov@dnbabkov laboratory]$ git add .
```

```
[dnbabkov@dnbabkov laboratory]$ git commit -a
[master 9a2afa6] Not first commit
 Committer: Dmitry Nikolaevich Babkov <dnbabkov@dnbabkov.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 455 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
```

```
Not first commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Committer: Dmitry Nikolaevich Babkov <dnbabkov@dnbabkov.localdomain
>
#
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   .gitignore
#       new file:   LICENSE
#
~
~
~
~
-- INSERT --
```

```
[dnbabkov@dnbabkov laboratory]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 5, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 6.45 KiB | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To git@github.com:dnbabkov/intro-os.git
   12f87e5..9a2afa6  master -> master
```

Конфигурация git-flow

- Устанавливаем git-flow для CentOS (Рис.15, 16)

```
[dnbabkov@dnbabkov laboratory]$ sudo yum install epel-release
[dnbabkov@dnbabkov laboratory]$ sudo yum install gitflow
```

- Инициализируем git-flow командой git flow init, префикс устанавливаем v (Рис.17)

```
[dnbabkov@dnbabkov laboratory]$ git flow init

Which branch should be used for bringing forth production releases?
   - master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

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

- Командой git branch проверяем, что мы находимся на ветке develop (Рис.18)

```
[dnbabkov@dnbabkov laboratory]$ git branch
* develop
  master
```

- Командой git flow release start 1.0.0 создаем релиз с версией 1.0.0 (Рис.19) и запишем версию (Рис.20)

```
[dnbabkov@dnbabkov laboratory]$ git flow release start 1.0.0
Switched to a new branch 'v1.0.0'

Summary of actions:
- A new branch 'v1.0.0' was created, based on 'develop'
- You are now on branch 'v1.0.0'

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'

[dnbabkov@dnbabkov laboratory]$ echo "1.0.0" >> VERSION
```

- Добавим в индекс командами git add . и git commit -am ‘chore(main): add version’ (Рис.21)

```
[dnbabkov@dnbabkov laboratory]$ git add .
[dnbabkov@dnbabkov laboratory]$ git commit -am 'chore(main): add version'
[v1.0.0 afbe83d] chore(main): add version
 Committer: Dmitry Nikolaevich Babkov <dnbabkov@dnbabkov.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 VERSION
```

- Залью релизную ветку в основную командой git flow release finish 1.0.0

```
[dnbabkov@dnbabkov laboratory]$ git flow release finish 1.0.0
Switched to branch 'master'
Merge made by the 'recursive' strategy.
 VERSION | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 VERSION
warning: refname 'v1.0.0' is ambiguous.
Switched to branch 'develop'
warning: refname 'v1.0.0' is ambiguous.
warning: refname 'v1.0.0' is ambiguous.
Merge made by the 'recursive' strategy.
 VERSION | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 VERSION
Deleted branch v1.0.0 (was afbe83d).

Summary of actions:
- Latest objects have been fetched from 'origin'
- Release branch has been merged into 'master'
- The release was tagged 'v1.0.0'
- Release branch has been back-merged into 'develop'
- Release branch 'v1.0.0' has been deleted
```

(Рис. 22)

- И последним пунктом мы отправим данные на github командами git

```
[dnbabkov@dnbabkov laboratory]$ git push --all
Counting objects: 6, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 489 bytes | 0 bytes/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local o
To git@github.com:dnbabkov/intro-os.git
   9a2afa6..be1b83c  master -> master
 * [new branch]      develop -> develop
```

push –all и git push –tags (Рис.23, 24)

```
[dnbabkov@dnbabkov laboratory]$ git push --tags
Counting objects: 1, done.
Writing objects: 100% (1/1), 173 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:dnbabkov/intro-os.git
 * [new tag]         v1.0.0 -> v1.0.0
```

Выводы:

Я изучил идеологию и применение средств контроля версий git и научился их использовать.

Контрольные вопросы: