

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО

ЛАБОРАТОРНОЙ РАБОТЕ №1

дисциплина: Научное программирование

Студент: Милёхин Александр

Группа: НПМмд-02-21

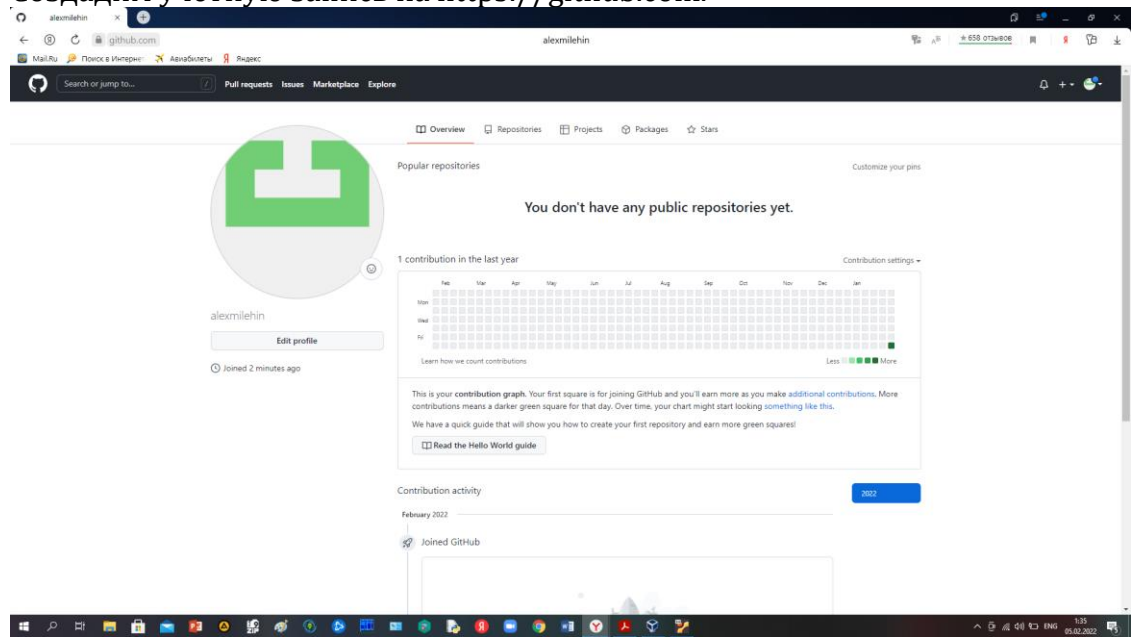
Ст. билет № 1032212345

Цель работы

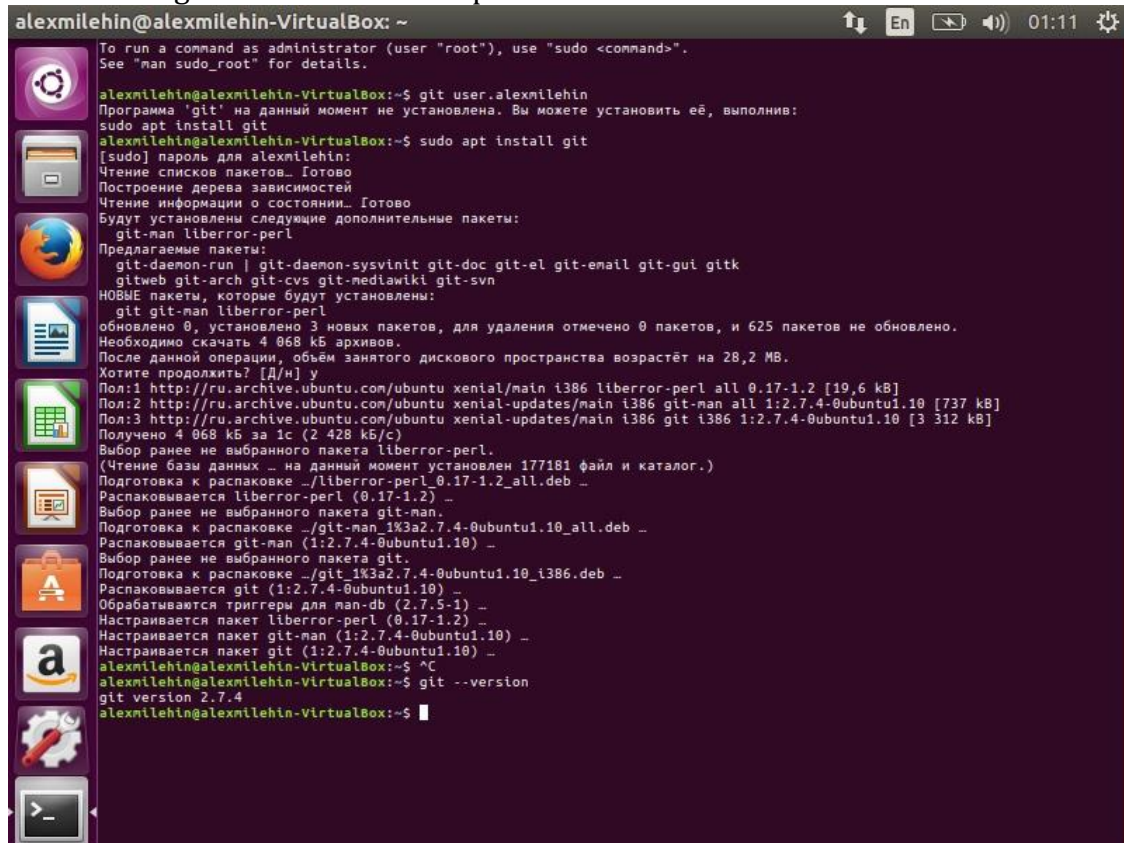
Изучить идеологию и применение средств контроля версий.

Ход работы

- 1) Создадим учётную запись на <https://github.com>.



2) Установим git на наш компьютер.



```
alexnmilehin@alexnmilehin-VirtualBox: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

alexnmilehin@alexnmilehin-VirtualBox:~$ git user.alexnmilehin
Программа 'git' на данный момент не установлена. Вы можете установить её, выполнив:
sudo apt install git
alexnmilehin@alexnmilehin-VirtualBox:~$ sudo apt install git
[sudo] пароль для alexnmilehin:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  git-man liberror-perl
Предлагаемые пакеты:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
НОВЫЕ пакеты, которые будут установлены:
  git git-man liberror-perl
обновлено 0, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 625 пакетов не обновлено.
Необходимо скачать 4 068 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 28,2 МБ.
Хотите продолжить? [Д/н] y
Пол:1 http://ru.archive.ubuntu.com/ubuntu xenial/main i386 liberror-perl all 0.17-1.2 [19,6 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu xenial-updates/main i386 git-man all 1:2.7.4-0ubuntu1.10 [737 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu xenial-updates/main i386 git i386 1:2.7.4-0ubuntu1.10 [3 312 kB]
Получено 4 068 кБ за 1с (2 428 кБ/с)
Выбор ранее не выбранного пакета liberror-perl.
(Чтение базы данных ... на данный момент установлен 177181 файл и каталог.)
Подготовка к распаковке ./liberror-perl_0.17-1.2_all.deb ...
Распаковывается liberror-perl (0.17-1.2) ...
Выбор ранее не выбранного пакета git-man.
Подготовка к распаковке ./git-man_1%3a2.7.4-0ubuntu1.10_all.deb ...
Распаковывается git-man (1:2.7.4-0ubuntu1.10) ...
Выбор ранее не выбранного пакета git.
Подготовка к распаковке ./git_1%3a2.7.4-0ubuntu1.10_i386.deb ...
Распаковывается git (1:2.7.4-0ubuntu1.10) ...
Обрабатываются триггеры для man-db (2.7.5-1) ...
Настраивается пакет liberror-perl (0.17-1.2) ...
Настраивается пакет git-man (1:2.7.4-0ubuntu1.10) ...
Настраивается пакет git (1:2.7.4-0ubuntu1.10) ...
alexnmilehin@alexnmilehin-VirtualBox:~$ ^C
alexnmilehin@alexnmilehin-VirtualBox:~$ git --version
git version 2.7.4
alexnmilehin@alexnmilehin-VirtualBox:~$
```

3) Настроим систему контроля версий git, как это указано в инструкции к 1-ой лабораторной работе с использованием сервера репозитория <https://github.com/>.

Для этого необходимо сгенерировать пару ключей (приватный и открытый), а затем вставить их в SSH-ключи на github.

```
Терминал
alexmilehin@alexmilehin-VirtualBox: ~
alexmilehin@alexmilehin-VirtualBox:~$ ssh-keygen -C"alexmilehin <1032212345@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alexmilehin/.ssh/id_rsa):
/home/alexmilehin/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alexmilehin/.ssh/id_rsa.
Your public key has been saved in /home/alexmilehin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:dXXGTdyDLxhey2cDOP7rvXDzeaLCztk2P8RbWGX+rzC alexmilehin <1032212345@pfur.ru>
The key's randomart image is:
+---[RSA 2048]---+
|          .o==B|
|         o.++*O|
|        o o..*=|
|       . o    .|=|
|      S  .. o.|
|         .+ o|
|        . o.=|
|       .ooo.BEB|
|      .+o+=oO=|
+---[SHA256]-----+
alexmilehin@alexmilehin-VirtualBox:~$
```

```
Терминал
alexmilehin@alexmilehin-VirtualBox: ~
alexmilehin@alexmilehin-VirtualBox:~$ sudo apt install xclip
[sudo] пароль для alexmilehin:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
НОВЫЕ пакеты, которые будут установлены:
  xclip
обновлено 0, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 625
пакетов не обновлено.
Необходимо скачать 16,5 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 67,6 к
Б.
Пол:1 http://ru.archive.ubuntu.com/ubuntu xenial/universe i386 xclip i386 0.12+s
vn84-4 [16,5 kB]
Получено 16,5 кБ за 0с (218 кБ/с)
Выбор ранее не выбранного пакета xclip.
(Чтение базы данных ... на данный момент установлено 177996 файлов и каталогов.)
Подготовка к распаковке .../xclip_0.12+svn84-4_i386.deb ...
Распаковывается xclip (0.12+svn84-4) ...
Обрабатываются триггеры для man-db (2.7.5-1) ...
Настраивается пакет xclip (0.12+svn84-4) ...
alexmilehin@alexmilehin-VirtualBox:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
alexmilehin@alexmilehin-VirtualBox:~$
```

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



alexmilehin

SHA256:dXXGTdyD1xhey2cDOP7rvXDzealCztk2P8RbWGX+rzc

Added on 5 Feb 2022

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

- Следующим шагом будет создание и подключение репозитория к GitHub. В GitHub заходим в «repository» и создаём новый репозиторий (имя «laboratory», а заголовок для файла README). Копируем в консоль ссылку на репозиторий (для дальнейшей работы с файлами):

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



alexmilehin ▾

Repository name *

/ laboratory ✓

Great repository names are short and memorable. Need inspiration? How about [bookish-winner?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



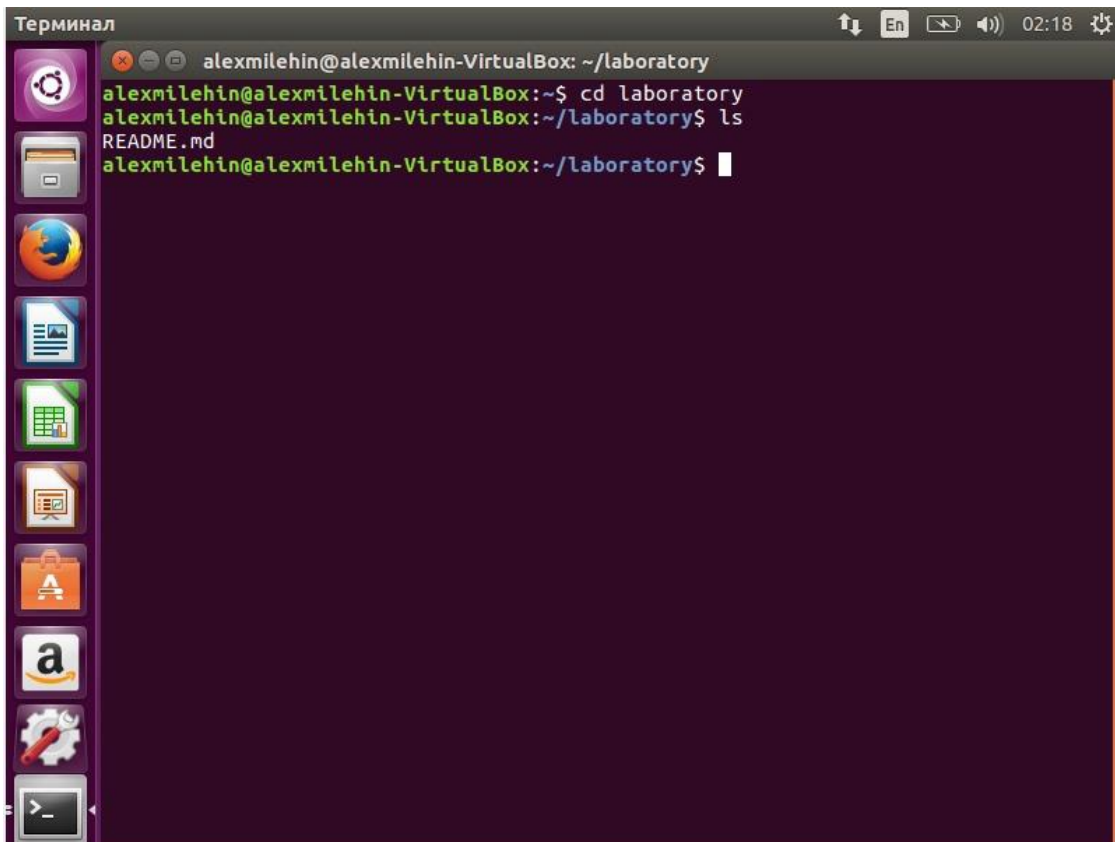
Private

You choose who can see and commit to this repository.

```
Терминал
alexmilehin@alexmilehin-VirtualBox: ~
alexmilehin@alexmilehin-VirtualBox:~$ git clone https://github.com/alexmilehin/laboratory.git
Клонирование в «laboratory»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Распаковка объектов: 100% (3/3), готово.
Проверка соединения... готово.
alexmilehin@alexmilehin-VirtualBox:~$
```

Рабочий каталог будем обозначать как `laboratory`. Вначале нужно перейти в этот каталог:

`-cd laboratory`



```
Терминал
alexmilehin@alexmilehin-VirtualBox: ~/laboratory
alexmilehin@alexmilehin-VirtualBox:~$ cd laboratory
alexmilehin@alexmilehin-VirtualBox:~/laboratory$ ls
README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory$
```

После этого можем создавать наши файлы.

Инициализируем системы git:

```
-git init
```

Создаём заготовку для файла README.md:

```
-echo "# Лабораторные работы" >> README.md
```

```
-git add README.md
```

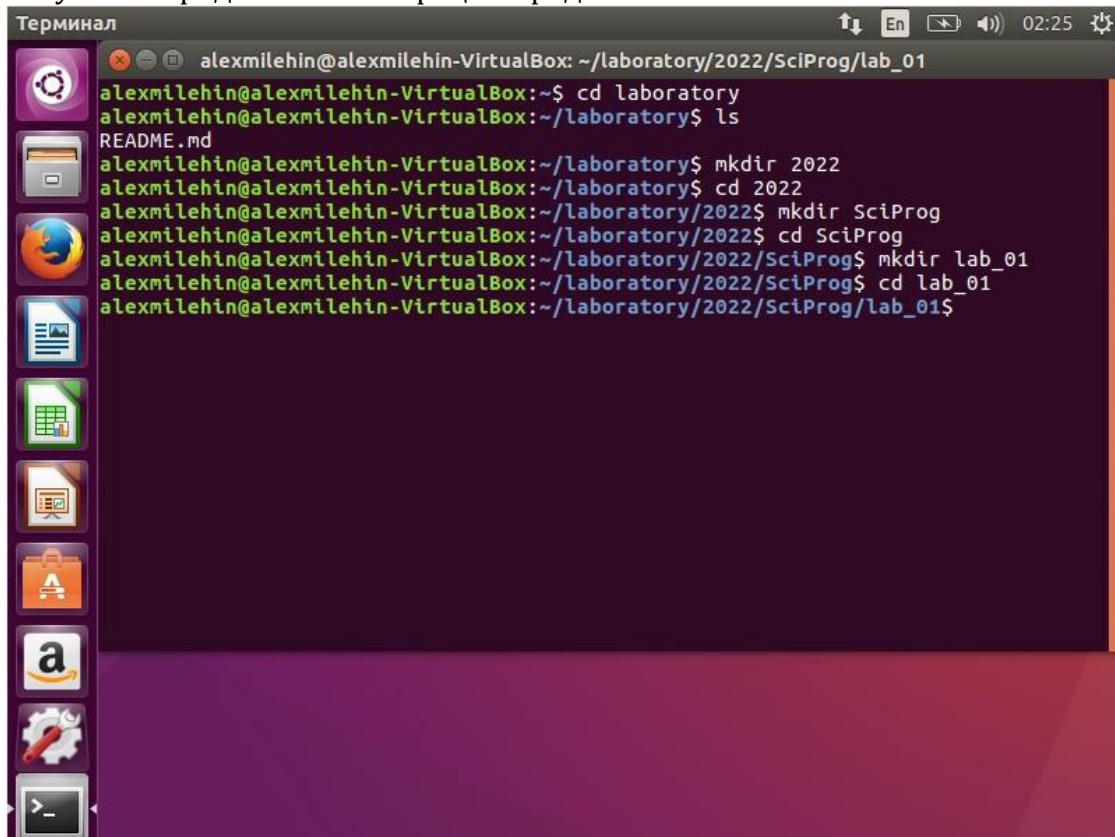
Делаем первый коммит и выкладываем на github:

```
-git commit -m "first commit"
```

```
-git remote add origin git@github.com:/sciproc-intro.git
```

```
-git push -u origin master
```

Результат проделанных операций представлен ниже.



The image shows a terminal window titled "Терминал" (Terminal) with a dark purple background. The window is running on a virtual machine named "alexmilehin-VirtualBox". The user "alexmilehin" is at the prompt. The terminal shows the following sequence of commands and output:

```
alexmilehin@alexmilehin-VirtualBox: ~/laboratory/2022/SciProg/lab_01
alexmilehin@alexmilehin-VirtualBox:~$ cd laboratory
alexmilehin@alexmilehin-VirtualBox:~/laboratory$ ls
README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory$ mkdir 2022
alexmilehin@alexmilehin-VirtualBox:~/laboratory$ cd 2022
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022$ mkdir SciProg
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022$ cd SciProg
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg$ mkdir lab_01
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg$ cd lab_01
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$
```

The terminal window has a title bar with standard Linux window controls (close, maximize, minimize) and system status icons (network, volume, battery) on the right. The left sidebar of the virtual machine desktop is visible, showing icons for various applications like a web browser, file manager, and terminal.

```
Терминал
alexmilehin@alexmilehin-VirtualBox: ~/laboratory/2022/SciProg/lab_01
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git init
Инициализирован пустой репозиторий Git в /home/alexmilehin/laboratory/2022/SciProg/lab_01/.git/
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ echo "# Лабораторные работы" >> README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git add README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git commit -m "first commit"

*** Please tell me who you are.

Run

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

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'alexmilehin@alexmilehin-VirtualBox.(none)')
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config --global --list
fatal: unable to read config file '/home/alexmilehin/.gitconfig': Нет такого файла или каталога
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true

alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ mkdir lab_01
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ cd lab_01
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git init
Инициализирован пустой репозиторий Git в /home/alexmilehin/laboratory/2022/SciProg/lab_01/.git/
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config --global user.email "am11999@yandex.ru"
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config --global user.name "alexmilehin"
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ echo "# Лабораторные работы" >> README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git add README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git commit -m "first commit"

*** Please tell me who you are.

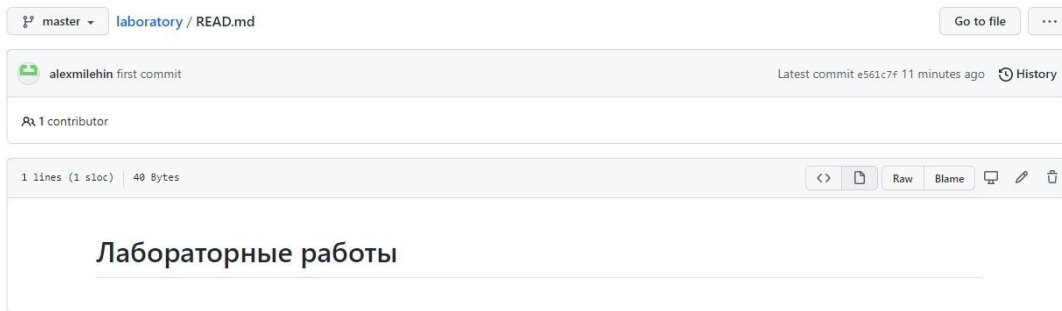
Run

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

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'alexmilehin@alexmilehin-VirtualBox.(none)')
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config user.email "am11999@yandex.ru"
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git config user.name "alexmilehin"
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git commit -m "first commit"
[master (корневой коммит) e561c7f] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git push -u origin master
fatal: 'origin' does not appear to be a git repository
fatal: could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git remote add origin git@github.com:alexmilehin/laboratory.git
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ git push -u origin master
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ECDSA key fingerprint is SHA256:p200H0IC17MeI0ttrv98R10uN03/1ykguFqN.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,140.82.121.4' (ECDSA) to the list of known hosts.
Посчет объектов: 3, готово.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:alexmilehin/laboratory.git
 * [new branch] master -> master
Ветка master отправлена на сервер ветку master на origin.
alexmilehin@alexmilehin-VirtualBox:~/laboratory/2022/SciProg/lab_01$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt
--2022-02-16 23:25:22-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознается creativecommons.org (creativecommons.org)... 104.20.151.16, 172.67.34.140, 104.20.150.16, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.151.16]:443... соединение установлено.
HTTP-запрос отправлен, ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в каталог: --legalcode.txt--
legalcode.txt [ <-- ] 18,22K ---KB/s in 0,001s
2022-02-16 23:25:23 (12,8 MB/s) - 'legalcode.txt' сохранён [18657]
--2022-02-16 23:25:23-- http://-/-/
Распознается (-).... ошибка: Имя или служба не известны.
wget: не удалось разрешить адрес: -
--2022-02-16 23:25:23-- http://o/
Распознается o (o).... 0.0.0.0
Подключение к o (o)[0.0.0.0]:80... ошибка: в соединении отказано.
```

6) Первичная конфигурация

Добавим файл лицензии:

```
-wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O
```

Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов: -
`curl -L -s https://www.gitignore.io/api/list`

Затем скачаем шаблон, например, для C:

```
-curl -L -s https://www.gitignore.io/api/c >> .gitignore
```

Можно это же сделать через web-интерфейс на сайте <https://www.gitignore.io/>.

Добавим новые файлы:

```
-git add . Выполним коммит:
```

```
-git commit -a Отправим на github:
```

```
-git push
```

Результат проделанных операций представлен ниже.

```
alexnieh@alexnieh-VirtualBox: /laboratory/2022/SclProg/lab_01$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O
--2022-02-16 23:25:22-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Подключение к creativecommons.org (creativecommons.org)... 104.20.151.16, 172.67.34.140, 104.20.150.16, ...
HTTP-запрос отправлен. Описание статуса... 200 OK
Длина: нет данных [text/plain]
Сохранение в каталог: "legalcode.txt"
legalcode.txt [ <== ] 18,22K ...KB/s ln 0,001s
2022-02-16 23:25:23 (12,8 MB/s) - "legalcode.txt" сохранён [18657]
--2022-02-16 23:25:23-- http://-
Распознаётся - (-)... ошибка: имя или служба не известны.
wget не удалось распознать адрес "-".
--2022-02-16 23:25:23-- http://0/
Распознаётся 0 (0)... 0.0.0.0
Подключение к 0 (0)[0.0.0.0]:80... ошибка: В соединении отказано.
ЗАВЕРШЕНО --2022-02-16 23:25:23--
Общее время: 1.45
Загружено: 1 файл, 18K за 0,001s (12,8 MB/s)
alexnieh@alexnieh-VirtualBox: /laboratory/2022/SclProg/lab_01$ sudo apt install curl
[sudo] пароль для alexnieh:
Чтение списков пакетов. Готово
Построение дерева зависимостей
Чтение информации о состоянии. Готово
Будут установлены следующие дополнительные пакеты:
libcurl3-gnutls
Новые пакеты, которые будут установлены:
curl
Пакеты, которые будут обновлены:
libcurl3-gnutls
обновлено 1; установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 474 пакетов не обновлено.
Необходимо скачать 353 Кб архивов.
После данной операции, объём занятого дискового пространства возрастёт на 350 Кб.
Хотите продолжить? [Y/n] y
Получено 353 Кб за 1с (200 КБ/с)
Чтение базы данных -- на данный момент установлено 178000 файлов и каталогов.
Подготовка к распаковке ./libcurl3-gnutls:7.47.0-1ubuntu2.19.1386.deb ...
Распаковывается libcurl3-gnutls:7.47.0-1ubuntu2.19 на замену (7.47.0-1ubuntu2.2) ...
Выбор ранее не выбранного пакета curl.
Подготовка к распаковке ./curl:7.47.0-1ubuntu2.19.1386.deb ...
Распаковывается curl (7.47.0-1ubuntu2.19) ...
Обрабатываются триггеры для libc-bin (2.23-0ubuntu9) ...
Настраивается пакет libcurl3-gnutls:7.47.0-1ubuntu2.19 ...
Настраивается пакет curl (7.47.0-1ubuntu2.19) ...
Обрабатываются триггеры для libc-bin (2.23-0ubuntu9) ...
alexnieh@alexnieh-VirtualBox: /laboratory/2022/SclProg/lab_01$ curl -L -S https://www.gitignore.io/api/list
1c,1c-bitrix,a-frame,actionscrip,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartus,altium,amplify,android,androidstudio
angular,anjuta,ansible,ansibletower,apachecordova
apachehadoop,appbuilder,appcelerator titanium,appcode,appcode+all
appcode+inl,appendengine,aptanastudio,arcanist,archive
archives,archlinuxpackages,aspnetcore,assembler,ate
atneltstudio,ats,audio,automationstudio,autotools
autotools+strict,awr,azurefunctions,azurite,backup
ballerina,basercms,basic,batch,bazaar
bazel,bitrise,bitrix,bittorrent,blackbox
bloop,bluej,bookdown,bower,bricxcc
buck,c,c++,cake,cakephp
cakephp2,cakephp3,calabash,carthage,certificates
ceylon,cfwheels,cheffcookbook,chocolatey,clean
clion,clion+all,clion+inl,clojure,cloud9
cnae,cocoapods,cocos2dx,cocoscreator,codeblocks
codecomposerstudio,codeigniter,codeio,codekit,codesniffer
coffeescript,commonlisp,compodoc,composer,compressed
compressedarchive,compression,conan,concrete5,coq
cordova,craftcms,crashlytics,crbasic,crossbar
crystal,cs-cart,csharp,cuda,cvs
cypressio,d,dart,darteditor,data
database,datarecovery,dbeaver,defold,delphi
dframe,diff,dlrenv,diskimage,django
dn,docfx,docpress,docz,dotenv
dotflessch,dotnetcore,dotsettings,dreamweaver,dropbox
drupal,drupal7,drupal8,e2studio,eagle
easybook,eclipse,eiffelstudio,elasticbeanstalk,elisp
elixir,elm,emacs,ember,engine
epi-server,erlang,espresso,executable,exercism
expressionengine,extjs,fancy,fastlane,finale
firebase,flashbuilder,flask,flatpak,flex
flexbuilder,floobits,flutter,font,fontforge
forcedotcom,forgegradle,fortran,freecad,freepascal
fsharp,fuelphp,fusetools,games,gcov
genero4gl,geth,ggts,gis,git
gitbook,go,godot,goland,goodsync
gpg,gradle,grails,greenfoot,groovy
grunt,gwt,haskell,helm,hexo
hol,homeassistant,homebrew,hsp,hugo
hyperledgercomposer,iar,iar_ewarn,larenbeddedworkbench,ldapro
idris,igorpro,images,infer,inforcms
inforcrm,intellij,intellij+all,intellij+inl,ionic3
jabref,java,jboss,jboss-4-2-3-ga,jboss-6-x
jboss4,jboss6,jdeveloper,jekyll,jenv
jetbrains,jetbrains+all,jetbrains+inl,jgiven,jigsaw
jmeter,joe,joomla,jspn,julia
jupyternotebooks,justcode,kaldi,kate,kdevelop4
kdiff3,kell,kentico,keystonejs,kicad
kirby2,kobalt,kohana,konodoeit,konyvisualizer
kotlin,labview,labviewnrg,lamp,laravel
latex,lazarus,leinigen,lemonstand,less
liberosoc,librarian-chef,libreoffice,lighthouse,illypond
linux,lithium,localstack,logtalk,lsspace
ltspace,lua,lyx,macos,magento
magento1,magento2,magic-xpa,matlab,maven
mavensnake,mdbook,mean,mercurial,mercury
```



```
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01
warning: push.default не установлен; его значение было изменено в Git версии 2.0 с «matching» на «single». Чтобы прекратить вывод этого сообщения и сохранить старое поведение, используйте:
git config --global push.default matching

Чтобы прекратить вывод этого сообщения и использовать новое поведение, используйте:
git config --global push.default single

Когда push.default установлено в «matching», git будет отправлять изменения локальных веток в существующие внешние ветки с таким же именем.
Начиная с Git версии 2.0, по умолчанию используется более консервативное поведение «single», которое отправляет изменения текущей ветки в соответствующую внешнюю ветку, из которой «git pull» забирает изменения.
Смотрите «git help config» и ищите «push.default» для дополнительной информации.
(режим «single» появился в Git версии 1.7.11. Используйте похожий режим «current» вместо «single», если вы иногда используете старые версии Git)

Warning: Permanently added the ECDSA host key for IP address '140.82.121.3' to the list of known hosts.
Подсчет объектов: 4, готово.
Delta compression using up to 2 threads.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 6.43 KiB | 0 bytes/s, готово.
Total 4 (delta 0), reused 0 (delta 0)
To git@github.com:alexmliehn/laboratory.git
e5d1c1ff..a47c135 master -> master
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git pull
Already up-to-date.
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ sudo apt-get install git-flow
[sudo] пароль для alexmliehn:
Чтение списков пакетов. Готово
Построение дерева зависимостей
Чтение информации о состоянии. Готово
С: Не удалось найти пакет git-flow
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ sudo apt-get install git-flow
Чтение списков пакетов. Готово
Построение дерева зависимостей
Чтение информации о состоянии. Готово
Новые пакеты, которые будут установлены:
git-flow
Обновлено 0, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 474 пакетов не обновлено.
Необходимо скачать 37,4 kB архивов.
После данной операции, объем занятого дискового пространства возрастет на 312 kB.
Плат: http://ru.archive.ubuntu.com/ubuntu xenial/universe 1386 git-flow all 1.9.1-1 [37,4 kB]
Получено 37,4 kB за 0с (94,8 kB/s).
Выбор ранее не выбранного пакета git-flow.
(Чтение базы данных ... на данный момент установлено 178015 файлов и каталогов.)
Подготовка к распаковке ./git-flow_1.9.1-1_all.deb ...
Распаковывается git-flow (1.9.1-1) ...
Настраивается пакет git-flow (1.9.1-1) ...
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow init
Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] alexmliehn
Fatal: Local branch 'alexmliehn' does not exist.
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ 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/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/alexmliehn/Laboratory/2022/ScIProg/lab_01/.git/hooks]
```

– Проверьте, что Вы на ветке develop:

git branch

```
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01
Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] alexmliehn
Fatal: Local branch 'alexmliehn' does not exist.
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ 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/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/alexmliehn/Laboratory/2022/ScIProg/lab_01/.git/hooks]
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
* develop
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow feature start feature_branch
Переходим на новую ветку «feature/feature_branch»
Summary of actions:
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'
Now, start committing on your feature. When done, use:
git flow feature finish feature_branch
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
* feature/feature_branch
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow feature finish feature_branch
Переходим на ветку «develop»
Already up-to-date.
Ветка feature/feature_branch удалена (бана a47c135).
Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
* develop
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow release start 1.0.0
Переходим 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'
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" > VERSION
1.0.0 > VERSION
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" >> VERSION
alexmliehn@alexmliehn-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git add
```

– Создадим релиз с версией 1.0.0

git flow release start 1.0.0

– Запишем версию:

echo “1.0.0” >> VERSION – Добавим в индекс:

git add .

git commit -am 'chore(main): add version'

```
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch

alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
develop
feature/feature_branch
master
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow feature finish feature_branch
Перехвачено на ветку «develop»
Already up-to-date.
Ветка feature/feature_branch удалена (SHA a47c135).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
develop
master
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow release start 1.0.0
Перехвачено 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'

alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" > VERSION
1.0.0 > VERSION
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" > VERSION
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git add .
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git commit -am 'chore(main): add version'
[release/1.0.0 a70d09e] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow release finish 1.0.0
Перехвачено на ветку «master»
Ваша ветка обновлена в соответствии с «origin/master».
Merge made by the 'recursive' strategy.
VERSION | 1
1 file changed, 1 insertion(+)
create mode 100644 VERSION
fatal: мет неопределен метка?
fatal: Tagging failed. Please run finish again to retry.
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git push --all
Pusher objects: 4, rotomo.
Delta compression using up to 2 threads.
Сжатие объектов: 100% (3/3), rotomo.
Замена объектов: 100% (4/4), 453 bytes | 0 bytes/s, rotomo.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To git@github.com:alexmtlehin/laboratory.git
+ a47c135..dc57cb3 master -> master
+ [new branch] develop -> develop
+ [new branch] release/1.0.0 -> release/1.0.0
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git push --tags
Everything up-to-date
alexmtlehin@alexmtlehin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$
```

– Зальём релизную ветку в основную ветку

git flow release finish 1.0.0

– Отправим данные на github

git push --all

git push --tags


```
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:

git flow feature finish feature_branch

alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
develop
* feature/feature_branch
master

alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow feature finish feature_branch
Переходим на ветку «develop»
Already up-to-date.
Ветка feature/feature_branch удалена (SHA a47c135).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git branch
develop
* develop
master

alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow release start 1.0.0
Переходим 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'

alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" > VERSION
1.0.0 > VERSION
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ echo "1.0.0" >> VERSION
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git add
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git commit -am 'chore(main): add version'
[release/1.0.0 a7ddc9e] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git flow release finish 1.0.0
Переходим на ветку «master».
Ветка master обновлена в соответствии с «origin/master».
Merge made by the 'recursive' strategy.
VERSION | 1
1 file changed, 1 insertion(+)
create mode 100644 VERSION
fatal: Tagging failed. Please run finish again to retry.
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git push --all
Посчет объектов: 4, готово.
Delta compression using up to 2 threads.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 451 bytes | 0 bytes/s, готово.
Total: 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:alexmlshin/laboratory.git
+47c135...d57903 master -> master
+ [new branch] develop -> develop
+ [new branch] release/1.0.0 -> release/1.0.0
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$ git push --tags
Everything up-to-date
alexmlshin@alexmlshin-VirtualBox: ~/Laboratory/2022/ScIProg/lab_01$
```

8. Создадим релиз на github. Для этого заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля (создаём теги для версии 1.0.0). После создания тега, автоматически сформируется релиз

Releases

Tags

Choose a tag

Target: master

Choose an existing tag, or create a new tag on publish

lab_01

Write

Preview

H

B

I

≡

<>

🔗

≡

≡

☑

@

📎

↶

+ Auto-generate release notes

Greetings!

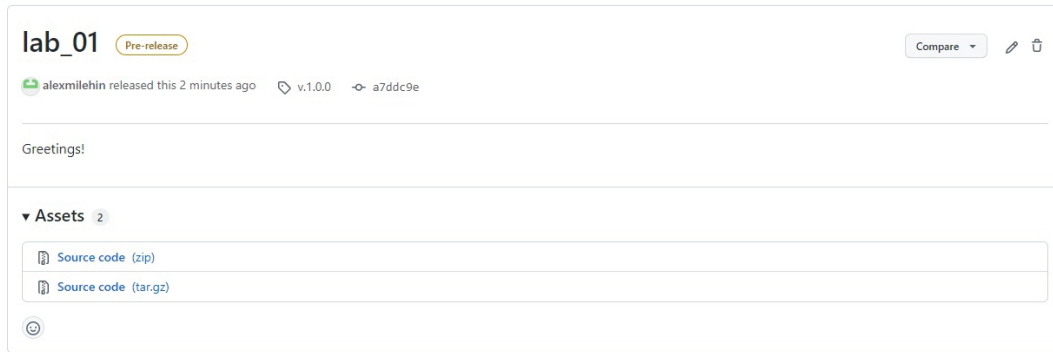
Attach files by dragging & dropping, selecting or pasting them.

↓ Attach binaries by dropping them here or selecting them.

☒ This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish release

Save draft



Выводы

Я изучил идеологию и научился применять средства контроля версий.

Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?
 - Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
 - Хранилище (repository), или репозиторий, — место хранения всех версий и служебной информации. Commit («[трудовой] вклад», не переводится) — синоним версии; процесс создания новой версии. Рабочая копия (working copy) — текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней).
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
 - Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществляется через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion. распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т. к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”. Еще пример - Wikipedia.

- В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов. Пример — Bitcoin.

- В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером.

4. Опишите действия с VCS при единоличной работе с хранилищем.

- Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория:

```
git config --global user.name "Имя Фамилия"
```

```
git config --global user.email "work@mail"
```

и настроив utf-8 в выводе сообщений git:

```
git config --global core.quotePath false
```

Для инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке:

```
cd
```

```
mkdir tutorial
```

```
cd tutorial
```

```
git init
```

5. Опишите порядок работы с общим хранилищем VCS.

- Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

```
ssh-keygen -C "Имя Фамилия work@mail"
```

Ключи сохраняются в каталоге ~/.ssh/.

Скопировав из локальной консоли ключ в буфер обмена

```
cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

вставляем ключ в появившееся на сайте поле.

6. Каковы основные задачи, решаемые инструментальным средством git?

- У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

7. Назовите и дайте краткую характеристику командам git.

- Наиболее часто используемые команды git:

– создание основного дерева репозитория:

`git init`

– получение обновлений (изменений) текущего дерева из центрального репозитория:
`git pull`

– отправка всех произведённых изменений локального дерева в центральный репозиторий:

`git push` – просмотр списка изменённых файлов в текущей директории: `git status`

– просмотр текущих изменения:

`git diff`

– добавить все изменённые и/или созданные файлы и/или каталоги: `git add .`

– добавить конкретные изменённые и/или созданные файлы и/или каталоги:

`git add имена_файлов` – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

`git rm имена_файлов`

– сохранение добавленных изменений:

– сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

– сохранить добавленные изменения с внесением комментария через встроенный редактор:

`git commit`

– создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

– переключение на некоторую ветку:

`git checkout имя_ветки`

– отправка изменений конкретной ветки в центральный репозиторий:

`git push origin имя_ветки`

– слияние ветки с текущим деревом:

`git merge --no-ff имя_ветки`

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

- Использование git при работе с локальными репозиториями (добавления текстового документа в локальный репозиторий):

```
git add hello.txt
```

```
git commit -am 'Новый файл'
```

9. Что такое и зачем могут быть нужны ветви (branches)?

- Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом. Кроме того, с помощью branches решаются следующие проблемы: нужно постоянно создавать архивы с рабочим кодом, сложно “переключаться” между архивами, сложно перетаскивать изменения между архивами, легко что-то напутать или потерять.

10. Как и зачем можно игнорировать некоторые файлы при commit?

- Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории. Во время работы над проектом эти файлы могут создаваться, но их не требуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл.gitignore с помощью сервисов. Для этого сначала нужно получить список имеющихся шаблонов:

```
curl -L -s https://www.gitignore.io/api/list
```

Затем скачать шаблон, например, для C и C++:

```
curl -L -s https://www.gitignore.io/api/c >> .gitignore
```

```
curl -L -s https://www.gitignore.io/api/c++ >> .gitignore
```