

Лабораторная работа 2. Markdown

Отчет по лабораторной работе 2

Хитяев Евгений Анатольевич НПИМд-02-21

Содержание

1	Цель работы	4
2	Теоретические сведения	5
3	Задание	6
4	Выполнение лабораторной работы	7
5	Выводы	16
6	Контрольные вопросы	17

List of Figures

4.1	Создание учетной записи	7
4.2	Установка git	8
4.3	Переход в каталог	10
4.4	Инициализация git-flow	13
4.5	Проверка	13
4.6	Создание релиза	14
4.7	Заливание ветки и отправка данных на github	14

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Теоретические сведения

Вся теоритическая часть по использованию языка разметки Markdown была взята из инструкции по лабораторной работе №2 на сайте: <https://esystem.rudn.ru/pluginfile.php/1284/markdown.pdf>

3 Задание

– Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. –
В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

4 Выполнение лабораторной работы

1. Создадим учётную запись на <https://github.com> (см. рис. 1)

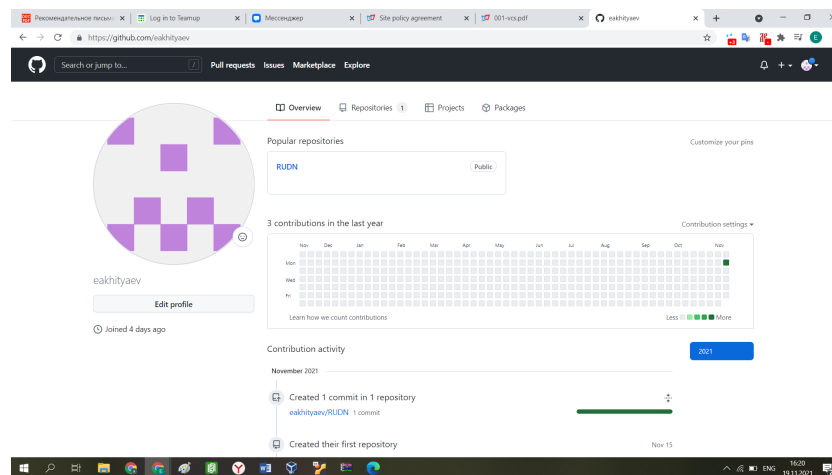
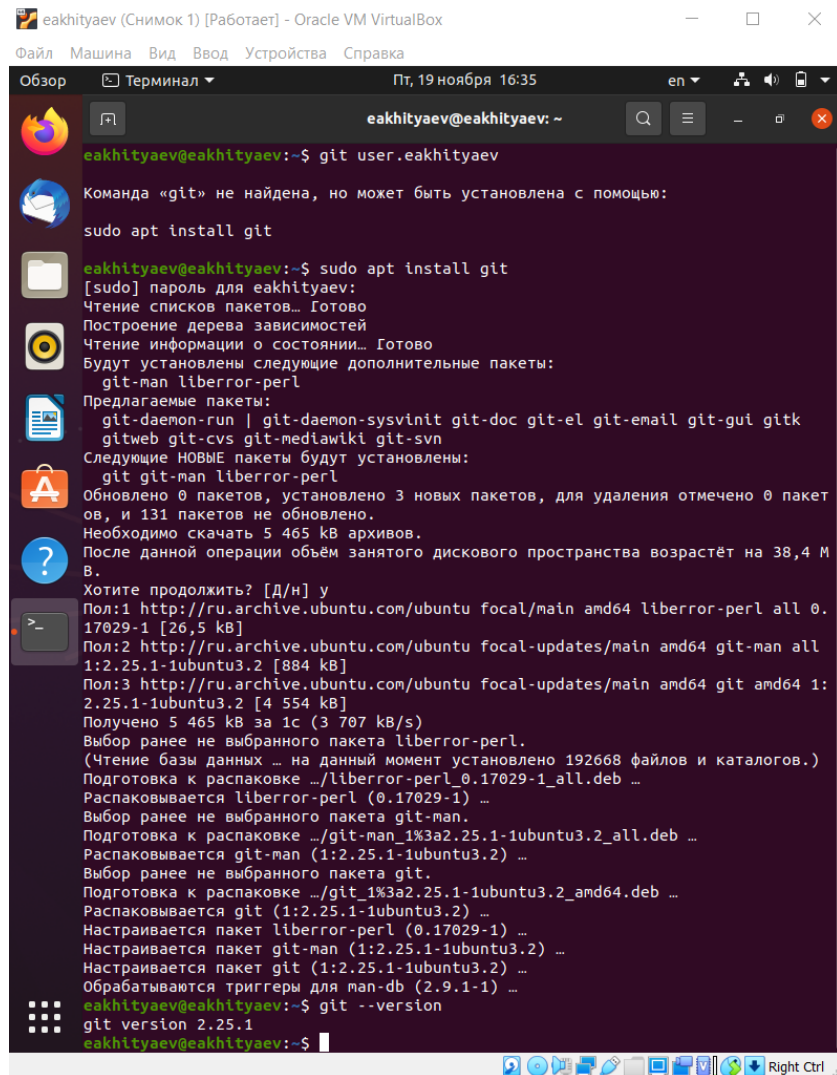


Figure 4.1: Создание учетной записи

2. Установим git на наш компьютер (алгоритм представлен на рис. 2).



```
eakhityaev@eakhityaev:~$ git user.eakhityaev
Команда «git» не найдена, но может быть установлена с помощью:
sudo apt install git

eakhityaev@eakhityaev:~$ sudo apt install git
[sudo] пароль для eakhityaev:
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-cvs git-mediawiki git-svn
Следующие НОВЫЕ пакеты будут установлены:
git git-man liberror-perl
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакет
ов, и 131 пакетов не обновлено.
Необходимо скачать 5 465 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 38,4 М
В.
Хотите продолжить? [Д/н] у
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.
17029-1 [26,5 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all
1:2.25.1-1ubuntu3.2 [884 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:
2.25.1-1ubuntu3.2 [4 554 kB]
Получено 5 465 kB за 1с (3 707 kB/s)
Выбор ранее не выбранного пакета liberror-perl.
(Чтение базы данных ... на данный момент установлено 192668 файлов и каталогов.)
Подготовка к распаковке ./liberror-perl_0.17029-1_all.deb ...
Распаковывается liberror-perl (0.17029-1) ...
Выбор ранее не выбранного пакета git-man.
Подготовка к распаковке ./git-man_1%3a2.25.1-1ubuntu3.2_all.deb ...
Распаковывается git-man (1:2.25.1-1ubuntu3.2) ...
Выбор ранее не выбранного пакета git.
Подготовка к распаковке ./git_1%3a2.25.1-1ubuntu3.2_amd64.deb ...
Распаковывается git (1:2.25.1-1ubuntu3.2) ...
Настраивается пакет liberror-perl (0.17029-1) ...
Настраивается пакет git-man (1:2.25.1-1ubuntu3.2) ...
Настраивается пакет git (1:2.25.1-1ubuntu3.2) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
eakhityaev@eakhityaev:~$ git --version
git version 2.25.1
eakhityaev@eakhityaev:~$
```

Figure 4.2: Установка git

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 (см. рис. 3, 4, 5).



```
eakhityaev (Снимок 1) [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Обзор Терминал Пт, 19 ноября 16:44 en
eakhityaev@eakhityaev: ~
eakhityaev@eakhityaev:~$ ssh-keygen -C"eakhityaev <1032212340@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/eakhityaev/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eakhityaev/.ssh/id_rsa
Your public key has been saved in /home/eakhityaev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VFgPzIssue7vwtUgBEti30c4l3rceNFoMPEPXC6+Dw eakhityaev <1032212340@pfur.ru>
The key's randomart image is:
+----[RSA 3072]-----+
... ++=
..o o.o B
..oo = * X +
..++B B =
..o.oS + .
..+.o
...
..o oE
..+o o.
+----[SHA256]-----+
eakhityaev@eakhityaev:~$
```

```
eakhityaev (Снимок 1) [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Обзор Терминал Пт, 19 ноября 16:5
eakhityaev@eakhityaev:~$ sudo apt install xclip
[sudo] пароль для eakhityaev:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
xclip
Обновлено 0 пакетов, установлено 1 новых пакетов,
и 131 пакетов не обновлено.
Необходимо скачать 18,4 кВ архивов.
После данной операции объем занятого дискового
В.
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal
-1 [18,4 kB]
Получено 18,4 кВ за 0с (46,3 кВ/с)
Выбор ранее не выбранного пакета xclip.
(Чтение базы данных ... на данный момент установл
Распаковывается xclip (0.13-1) ...
Настраивается пакет xclip (0.13-1) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
eakhityaev@eakhityaev:~$ cat ~/.ssh/id_rsa.pub
eakhityaev@eakhityaev:~$
```

SSH keys

New SSH key

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



SSH_EA
SHA256:GfE7/12wOGvrIiEZTeRQyHHSBsFZg1as6pp2XPxd9zk
Added on 19 Nov 2021
Never used — Read/write

Delete

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

4. 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) (см. рис. 6, 7). Копируем в консоль ссылку на репозиторий (для дальнейшей работы с файлами):

Search or jump to...

Pull requests Issues Marketplace Explore

Create a new repository

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

Owner *

Repository name *

eakhityaev / laboratory

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

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.

```
eakhityaev@eakhityaev:~$ git clone https://github.
Клонирование в «laboratory»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), ра
Распаковка объектов: 100% (3/3), 595 байтов | 59.6
eakhityaev@eakhityaev:~$
```

Рабочий каталог будем обозначать как laboratory. Вначале нужно перейти в

этот каталог (рис. 8): `–cd laboratory`

```
eakhityaev@eakhityaev:~$ cd laboratory
eakhityaev@eakhityaev:~/laboratory$ ls
README.md
eakhityaev@eakhityaev:~/laboratory$
```

Figure 4.3: Переход в каталог

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 (рис. 9, 10, 11, 12).

```
eakhityaev@eakhityaev:~/laboratory$ mkdir 2021-2022
eakhityaev@eakhityaev:~/laboratory$ cd 2021-2022
eakhityaev@eakhityaev:~/laboratory/2021-2022$ mkdir SciProg
eakhityaev@eakhityaev:~/laboratory/2021-2022$ cd SciProg
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ mkdir lab01
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ cd lab01
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg/lab01$
```

```
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git init
Переинициализирован существующий репозиторий Git в
021-2022/SciProg/lab01/.git/
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ echo "# Лабораторные работы" >> README.md
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git add README.md
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git commit -m "first commit"

*** Пожалуйста, скажите мне кто вы есть.

Запустите

git config --global user.email "you@example.com"
git config --global user.name "Ваше Имя"

для указания идентификационных данных аккаунта по
Пропустите параметр --global для указания данных т

fatal: не удалось выполнить автоопределение адреса
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git config --list
fatal: unable to read config file '/home/eakhityaev/.gitconfig'
или каталога
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git commit -m "first commit"
l "1032212340@pfur.ru"
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git push -u origin master
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$
"eakhityaev"
eakhityaev@eakhityaev:~/laboratory/2021-2022/SciProg$ git commit -m "first commit"
[master (корневой коммит) 6129a2a] first commit
1 file changed, 2 insertions(+)
create mode 100644 README.md
```

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$ git push -u origin master
Username for 'https://github.com': eakhityaev
Password for 'https://github.com': eakhityaev
remote: Permission to eakhityaev/laboratory.git denied to eakhityaev.
fatal: «https://github.com/eakhityaev/laboratory.git/» недоступно: The requested URL returned
error: 403
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$ git push -u origin master
Username for 'https://github.com': eakhityaev
Password for 'https://github.com': eakhityaev
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 247 байтов | 247.00 КиБ/с, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/eakhityaev/laboratory/pull/new/master
remote:
To https://github.com/eakhityaev/laboratory.git
 * [new branch]      master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$
```

laboratory / README.md

eakhityaev Update README.md

1 contributor

<> Raw Blame 1 lines (1 slot)

Лабораторные работы

5. Первичная конфигурация Добавим файл лицензии: `–wget https://creativecommons.org/licenses/by/4.0/legalcode.txt`
 –О 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 (рис. 13, 14, 15, 16, 17).

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O
--2021-11-20 00:59:53-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся 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	за 0,004s
2021-11-20 00:59:54 (4,89 MB/s) - «legalcode.txt» сохранён [18657]				

```
--2021-11-20 00:59:54-- http://
Распознаётся - (-)... ошибка: Неизвестное имя или служба.
wget: не удастся разрешить адрес «-»
--2021-11-20 00:59:54-- http://0/
Распознаётся 0 (0)... 0.0.0.0
Подключение к 0 (0)[0.0.0.0]:80... ошибка: В соединении отказано.
ЗАВЕРШЕНО --2021-11-20 00:59:54--
Общее время: 1,0s
Загружено: 1 файл, 18K за 0,004s (4,89 MB/s)
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$ curl -L -S https://www.gitignore.io/api/list

Команда «curl» не найдена, но может быть установлена с помощью:

sudo apt install curl
```

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$ [sudo] пароль для eakhityaev:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libcurl4
Следующие НОВЫЕ пакеты будут установлены:
  curl
Следующие пакеты будут обновлены:
  libcurl4
Обновлено 1 пакетов, установлено 1 новых пакетов, для удаления ничего не требуется.
Необходимо скачать 161 kB/396 kB архивов.
После данной операции объем занятого дискового пространства станет 396 kB.
Хотите продолжить? [д/н] y
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl 7.78.0-1ubuntu2.7 [161 kB]
Получено 161 kB за 1с (147 kB/s)
Чтение базы данных ... на данный момент установлено 193615 файлов в базе данных
Подготовка к распаковке ./libcurl4_7.68.0-1ubuntu2.7_amd64.deb ...
Распаковывается libcurl4:amd64 (7.68.0-1ubuntu2.7) на замену ранее установленному пакету libcurl4.
Выбор ранее не выбранного пакета curl.
Подготовка к распаковке ./curl_7.68.0-1ubuntu2.7_amd64.deb ...
Распаковывается curl (7.68.0-1ubuntu2.7) ...
Настраивается пакет libcurl4:amd64 (7.68.0-1ubuntu2.7) ...
Настраивается пакет curl (7.68.0-1ubuntu2.7) ...
Обработываются триггеры для man-db (2.9.1-1) ...
Обработываются триггеры для libc-bin (2.31-0ubuntu9.2) ...
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/Lab01$
```

```

eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ curl -L -S https://www.gitignore.io/api/list
1c,1c-bitrix,a-frame,actionsript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,ansibletower,apachecordova
apachehadoop,appbuilder,appcelerator titanium,appcode,appcode+all
appcode+tml,appengine,aptanastudio,arcanist,archive
archives,archlinuxpackages,aspnetcore,assembler,ate
atmelstudio,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,chefcookbook,chocolatey,clean
clion,clion+all,clion+tml,closure,cloud9
cnahe,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,direnv,diskimage,django
dm,docfx,docpress,docz,dotenv
dotflessch,dotnetcore,dotsettings,dreamweaver,dropbox
drupal,drupal7,drupal8,e2studio,eagle
easybook,eclipse,eiffelstudio,elasticsearch,elisp
elixir,elm,emacs,ember,enzyme
epi-server,erlang,espresso,executable,exercism
expressionengine,extjs,fancy,fastlane,finale
firebase,flashbuilder,flask,flatpak,flex
flexbuilder,floobits,flutter,font,fontforge
forcedotcom,forgegradle,fortran,freepascal,fsharp
fuelphp,fusetools,games,gcov,genero4gl
geth,ggts,gis,git,gitbook
go,godot,goland,goodsync,gpg
gradle,grails,greenfoot,groovy,grunt
grwt,haskell,helm,hexo,hol

eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git push
Username for 'https://github.com': eakhityaev
Password for 'https://eakhityaev@github.com':
Перечисление объектов: 8, готово.
Подсчет объектов: 100% (8/8), готово.
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 6.77 Киб | 2.25 Миб/с, готово.
Всего 6 (изменения 0), повторно использовано 0 (изменения 0)
To https://github.com/eakhityaev/laboratory.git
b23f08a..c4f893a master -> master
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$

```

```

Zukencr8000@eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ cu
i/c >> .gitignore
% Total % Received % Xferd Average Speed Time Time Time
Dload Upload Total Spent Left S
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00
100 631 100 631 0 0 609 0 0:00:01 0:00:01 0:00:00
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git add .
[master a725dbd] Hello, sciprog
2 files changed, 455 insertions(+)
create mode 100644 .gitignore
create mode 100644 legalcode.txt
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git push
Username for 'https://github.com': eakhityaev
Password for 'https://eakhityaev@github.com':
To https://github.com/eakhityaev/laboratory.git
1 [rejected] master -> master (fetch first)
error: не удалось отправить некоторые ссылки в https://github.com/eakhityaev/laboratory.git
подсказка: Обновления были отклонены, так как внешний репозиторий содержит
подсказка: изменения, которых у вас нет в вашем локальном репозитории.
подсказка: Обычно, это связано с тем, что кто-то уже отправил изменения
подсказка: то же место. Перед повторной отправкой ваших изменений, вам ну
подсказка: забрать и слить изменения из внешнего репозитория себе
подсказка: (например, с помощью «git pull ...»).
подсказка: Для дополнительной информации, смотрите «Note about fast-forward»
подсказка: в «git push --help».
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Распаковка объектов: 100% (6/6), 1.21 Киб | 1.21 Миб/с, готово.
Из https://github.com/eakhityaev/laboratory
* 6129a2a..b23f08a master -> origin/master
* [Новая ветка] main -> origin/main
Merge made by the 'recursive' strategy.
 README.md | 1
 1 file changed, 1 deletion(-)

```

6. Работа с конфигурацией git-flow. Инициализируем git-flow (рис. 18): git flow init
 Префикс для ярлычков установим в v.

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ sudo apt-get install git-flow
[sudo] пароль для eakhityaev:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  git-flow
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 130 пакетов не обновлено
Необходимо скачать 38,8 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 330 kB.
Полн: http://ru.archive.ubuntu.com/ubuntu focal/universe amd64 git-flow all 1.12.3-1 [38,8 kB]
Получено 38,8 kB за 1с (52,9 kB/s)
Выбор ранее не выбранного пакета git-flow.
(Чтение базы данных ... на данный момент установлено 193622 файла и каталога.)
Подготовка к распаковке ./git-flow_1.12.3-1_all.deb ...
Распаковывается git-flow (1.12.3-1) ...
Настраивается пакет git-flow (1.12.3-1) ...
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git flow init

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

Which branch should be used for integration of the "next release"?
- develop
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/eakhityaev/Laboratory/2021-2022/SciProg/lab01/.git/hooks]
```

Figure 4.4: Инициализация git-flow

Проверим, что Вы на ветке develop (рис. 19): git branch

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git branch
  develop
* master
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git flow feature start feature_branch
Переключено 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

eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git branch
  develop
* feature/feature_branch
  master
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git flow feature finish feature_branch
Переключено на ветку «develop»
Уже обновлено.
Ветка feature/feature_branch удалена (была c4f893a).

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'

eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git branch
  develop
* master
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$
```

Figure 4.5: Проверка

Создадим релиз с версией 1.0.0 (рис. 20): 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'

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ 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'

eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ echo "1.0.0" >> VERSION
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git add .
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git commit -am 'chore(main): add version'
[release/1.0.0 e45fa27] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
```

Figure 4.6: Создание релиза

Зальём релизную ветку в основную ветку (рис. 21): `git flow release finish 1.0.0`

Отправим данные на github: `git push --all git push --tags`

```
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git flow release finish 1.0.0
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Уже на «master»
Ваша ветка опережает «origin/master» на 3 коммита.
(Используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
fatal: Tagging failed. Please run finish again to retry.
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git push --all
Username for 'https://github.com': eakhityaev
Password for 'https://eakhityaev@github.com':
Перечисление объектов: 100% (8/8), готово.
Подсчет объектов: 100% (8/8), готово.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 613 байтов | 613.00 КиБ/с, готово.
Всего 7 (изменения 3), повторно использовано 0 (изменения 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/eakhityaev/Laboratory.git
 c4f893a..f8fa0df master -> master
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$ git push --tags
Username for 'https://github.com': eakhityaev
Password for 'https://eakhityaev@github.com':
Everything up-to-date
eakhityaev@eakhityaev:~/Laboratory/2021-2022/SciProg/lab01$
```

Figure 4.7: Заливание ветки и отправка данных на github

7. 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) (рис. 22). После создания тега, автоматически сформируется релиз (рис. 23).

eakhityaev / laboratoryPublic

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

ReleasesTags

v1.1

Target: release/1.0.0

Excellent! This tag will be created from the target when you publish this release.

lab01

Write

Preview

H B I

≡ <> ↻

☑ @ ↶ ↷

+ Auto-generate release notes

Hello!

eakhityaev / laboratoryPublic

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Releases / v1.1

lab01

eakhityaev

 released this now

v1.1

e45fa27

Hello!

▼ Assets 2

Source code (zip)

Source code (tar.gz)

15

5 Выводы

Изучил идеологию и научился применять средства контроля версий.

6 Контрольные вопросы

1) Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

- 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 осуществлялся 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 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 изменений:

```
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.

- Ипользования `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
```