

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:

Ездаков Егор Андреевич

Группа:

НПМбд-01-20

МОСКВА

2021 г.

Цель работы: изучить идеологию и применение средств контроля версий.

Ход работы:

1. Создал учетную запись на GitHub:

The screenshot shows a GitHub profile for user 'ezdakovea'. At the top, there's a navigation bar with 'Overview', 'Repositories', 'Projects', and 'Packages'. Below this is a profile picture placeholder (a yellow 'M' on a grey circle) and a bio section with a 'ProTip!' message and an 'Edit profile' button. The 'Popular repositories' section is empty, displaying 'You don't have any public repositories yet.' Below the profile picture, it says 'Joined 12 minutes ago'. To the right, there's a '1 contribution in the last year' section with a calendar grid showing contributions for May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, Jan, Feb, Mar, and Apr. The grid shows contributions for Mon and Wed in May.

2. Настроил систему контроля версий git:

1) Сгенерировал ssh-ключ:

```
Файл Правка Вид Поиск Терминал Справка
The key's randomart image is:
+---[RSA 2048]-----+
|
| ..
| ....
| o ++
| o+.o+
| o . S*+
| . . o&o+
| + . X=0+
| + B=B..+ .
| o+E++ .+
+---[SHA256]-----+
[ezdakov@ezdakov ~]$ cat /home/ezdakov/.ssh/id_rsa.pub
bash: cat /home/ezdakov/.ssh/id_rsa.pub: Нет такого файла или каталога
[ezdakov@ezdakov ~]$ cat /home/ezdakov/.ssh/id_rsa.pub
bash: cat /home/ezdakov/.ssh/id_rsa.pub: Нет такого файла или каталога
[ezdakov@ezdakov ~]$ cat /home/ezdakov/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAzDWIqFk01yh8PYztVzE8vcfEhI0fvAaWQiHpuB8B0
qj1tZpVhQMKX0ogS70kW0AZhCm/thjoPRsPA8bZ/mM9UNFAETlHLu+GK+AtbuMqyS+nbFoacfYJuBQn
rx8bX/CMFNeN0afftPFqGlgEEFGZ32HHAzzZmQ9anZpwoLBhnNPdouBLp6/9LaF3TNb5yKXFEUCXKoE
eYAJPPeDjCQqKc0bcmgTzDQm1VwSEMPmR+SGzs6/Pj1KlsUf87NyIXtbZgvxKD70+C3C0nMzY+4vnBUE
oUTNBKZKmSAB38+pLvmx23mt3zIXZqaerfJvoVwqSwgk2ZKi/ZnFESVdK1sL ezdakovea <ezdakov0
2@mail.ru>
[ezdakov@ezdakov ~]$ █
```

2) Добавил его в настройках учетной записи на GitHub:

SSH keys

New SSH key

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

The screenshot shows the 'SSH keys' section in GitHub settings. It displays a single SSH key: 'ssh-rsa AAAAB3NzaC1yc2EAAA...'. Below the key name, it shows the full key string: 'SHA256:p89IJYR619to3j0njFBE0qI+83tRPhDvmh5ywU/qm7M'. It also shows the date 'Added on May 1, 2021' and the status 'Never used — Read/write'. A 'Delete' button is visible next to the key.

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

3. Создал структуру каталога лабораторных работ с помощью mkdir:



```
[eaezdakov@eaezdakov ~]$ mkdir -p ~/work/2020-2021/Operating systems/laboratory  
[eaezdakov@eaezdakov ~]$
```

4. Создал репозиторий на GitHub. Назвал его os-intro:

Create a new repository


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

Owner * **Repository name ***

 ezdakovea ▾ / 

Great repository names are The repository os-intro already exists on this account. about p:

Description (optional)

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

5. Перешёл в каталог laboratory: cd

6. Инициализировал системы git:

```
[eaezdakov@eaezdakov laboratory]$ git init
```

7. Создал заготовку для файла README.md:

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

8. Создал первый коммит и отправил его на GitHub:

```
[eaezdakov@eaezdakov laboratory]$ echo "# Лабораторные работы" >> README.md
[eaezdakov@eaezdakov laboratory]$ git add README.md
[eaezdakov@eaezdakov laboratory]$ git commit -m "first commit"
[master (root-commit) 80b058e] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

[eaezdakov@eaezdakov laboratory]$ git remote add origin git@github.com:ezdakovea/os-intro.git
[eaezdakov@eaezdakov laboratory]$ git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
RSA key fingerprint is MD5:16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,140.82.121.3' (RSA) to the list of known hosts.
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:ezdakovea/os-intro.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
[eaezdakov@eaezdakov laboratory]$
```

9. Добавил файл лицензии:

```
[eaezdakov@eaezdakov laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-05-01 12:37:48-- 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... соединено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

[ <=> ] 18 657 ---K/s за 0,003s

2021-05-01 12:37:48 (6,28 MB/s) - «LICENSE» сохранён [18657]

[eaezdakov@eaezdakov laboratory]$
```

10. Просмотрел список имеющихся шаблонов игнорируемых файлов:

```
Файл  Правка  Вид  Поиск  Терминал  Справка
[eaездakov@eaездakov laboratory]$ curl -L -s https://www.gitignore.io/api/list
lc,lc-bitrix,a-frame,actionscript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,apachecordova,apachehadoop
appbuilder,appcelerator titanium,appcode,appcode+all,appcode+iml
appengine,aptanastudio,arcanist,archive,archives
archlinuxpackages,aspnetcore,assembler,ate,atmelstudio
ats,audio,automationstudio,autotools,autotools+strict
awr,azurefunctions,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+iml,clojure,cloud9,cmake,cocoapods
cocos2dx,cocoscreator,code,code-java,codeblocks
codecomposerstudio,codeigniter,codeio,codekit,codesniffer
coffeescript,commonlisp,compodoc,composer,compressed
```

11. Загрузил шаблон для C:

```
[eaездakov@eaездakov laboratory]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
[eaездakov@eaездakov laboratory]$
```

12. Добавил новые файлы, отправил на GutHub:

```
[eaездakov@eaездakov laboratory]$ git add .
[eaездakov@eaездakov laboratory]$ git commit -am 'Создали шаблон для C'
[master 7684635] Создали шаблон для C
 2 files changed, 514 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
```

```
[eaездakov@eaездakov laboratory]$ git push
```

13. Инициализировал git-flow:

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

14. Префикс для ярлыков установил в v:

```
Version tag prefix? [] v
```

15. Находясь на ветке develop, создал релиз с версией 1.0.0:

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

[eaezdakov@eaezdakov laboratory]$ git flow release start 1.0.0
Switched to a new branch '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'
```

16. Записал версию:

```
[eaezdakov@eaezdakov laboratory]$ echo "1.0.0" >> VERSION
[eaezdakov@eaezdakov laboratory]$ █
```

17. Добавил в индекс:

```
[eaezdakov@eaezdakov laboratory]$ git add .
[eaezdakov@eaezdakov laboratory]$ git commit -am 'chore(main): add version'
[release/1.0.0 b6644e1] chore(main): add version
 1 file changed, 1 insertion(+)
 create mode 100644 VERSION
```

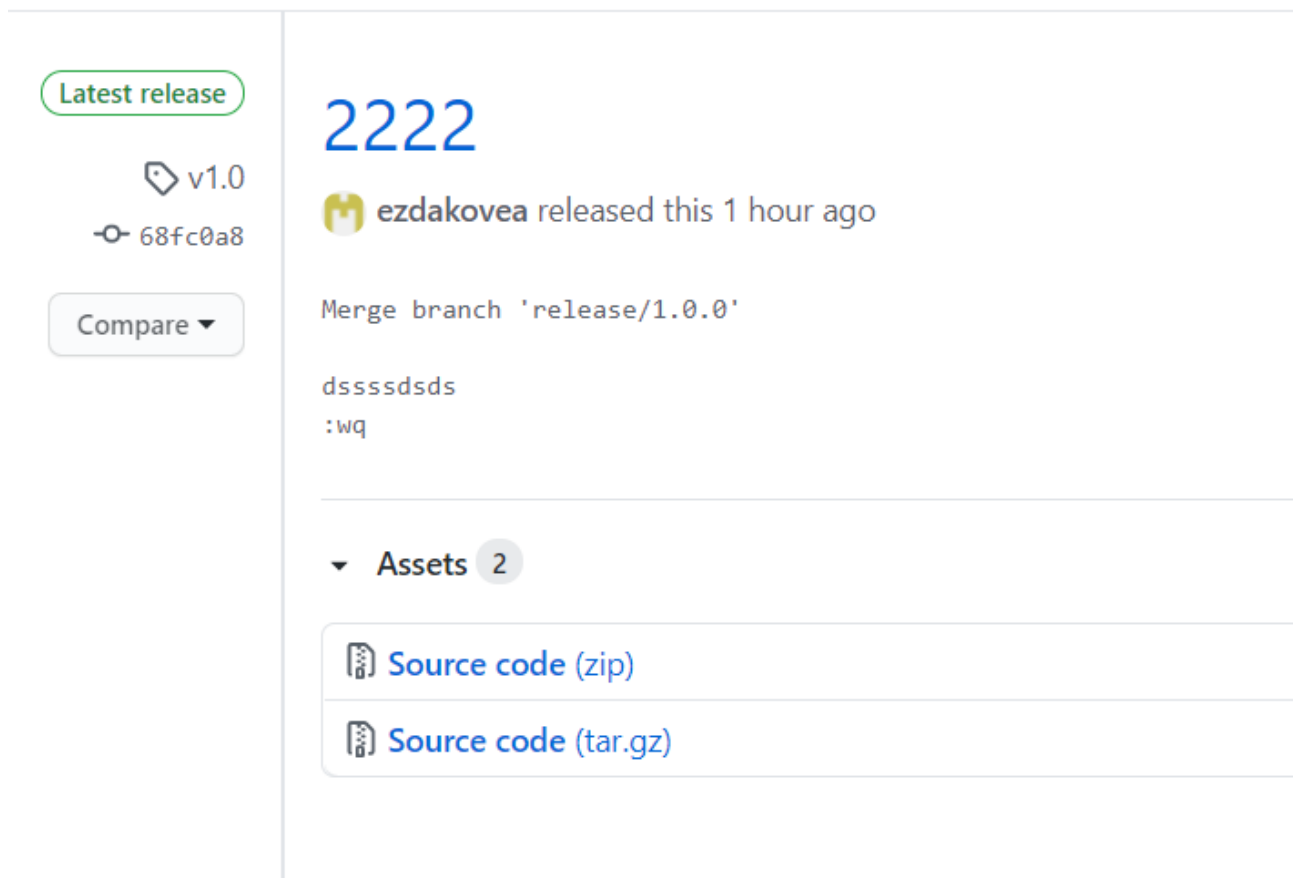
18. Залил релизную ветку в основную ветку:

```
----- merge -----
[eaezdakov@eaezdakov laboratory]$ git flow release finish 1.0.0
```

19. Отправил данные на GitHub:

```
[eaezdakov@eaezdakov laboratory]$ git push --all
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 386 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To git@github.com:ezdakovea/os-intro.git
 7684635..68fc0a8  master -> master
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
[eaezdakov@eaezdakov laboratory]$ git push --tags
Everything up-to-date
[eaezdakov@eaezdakov laboratory]$
```

20. Создаем 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.



Вывод: я изучил идеологию и применение средств контроля версий.

Ответы на контрольные вопросы:

1. Система контроля версий (VCS) – ПО, облегчающее реализацию удалённой работы нескольких пользователей над одним проектом.
2. Хранилище – это буквально хранилище для всей информации, которая используется в проекте. Commit – команда, позволяющая сохранить все добавленные изменения. История – сортированный по времени список всех когда-либо внесённых изменений. Рабочая копия – копия удалённого репозитория, расположенная на устройстве пользователя.
3. Централизованная (классическая) VCS представляет собой один репозиторий для хранения файлов. Самые известные примеры: CVS, Subversion. Децентрализованная (распределённая) VCS не предполагает обязательного наличия центрального репозитория, её суть в том, что файлы хранятся не на одном сервере, а на устройствах всех участников проекта. Примеры: Git, Bazaar, Mercurial.

4. При единоличной работе с хранилищем на устройстве создаётся локальный репозиторий, в файлы вносятся необходимые изменения, которые через commit отправляются уже в удалённый репозиторий.

5. При работе с общим хранилищем обычно файлы скачиваются из удалённого репозитория, создаётся новая ветка, на которой в файлы так же вносятся и сохраняются с помощью commit нужные изменения. После этого ветка отправляется в удалённый репозиторий, где сливается с основной веткой.

6. Основные задачи git – обеспечение удобства работы с версиями и хранения информации

7. – Наиболее часто используемые команды git: – создание основного дерева репозитория: git init – получение обновлений 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 – сохранение текущих изменений: – 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 (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) – отправка изменений конкретной ветки в центральный репозиторий: git push origin имя_ветки – слияние ветки стекущим деревом: git merge --no-ff имя_ветки – удаление ветки: – удаление локальной уже слитой с основным деревом ветки: git branch -d имя_ветки – принудительное удаление локальной ветки: git branch -D имя_ветки – удаление ветки с центрального репозитория: git push origin :имя_ветки

8. С локальным репозитием git можно использовать для работы с личными проектами, - система контроля версий позволяет удобно систематизировать файлы и экспериментировать, ведь в случае неудачи можно откатиться на более раннюю

версию. С удалённым репозиторием удобно осуществлять групповые работы, - помимо возможного отката на ранние версии здесь можно следить за работой каждого из участников, а значит, эффективнее распределять задачи и корректировать проект.

9. Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом.

10. Некоторые файлы не нужно добавлять в проект, - например, файлы редакторов, остаточные файлы и прочий мусор. С помощью `gitignore` можно удобно выбрать шаблон игнорирования и автоматически избавить себя от необходимости «чистить» репозиторий от ненужных файлов.