

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: Куликова Юлия Викторовна

Группа: НПМбв-01-18

МОСКВА

2022 г.

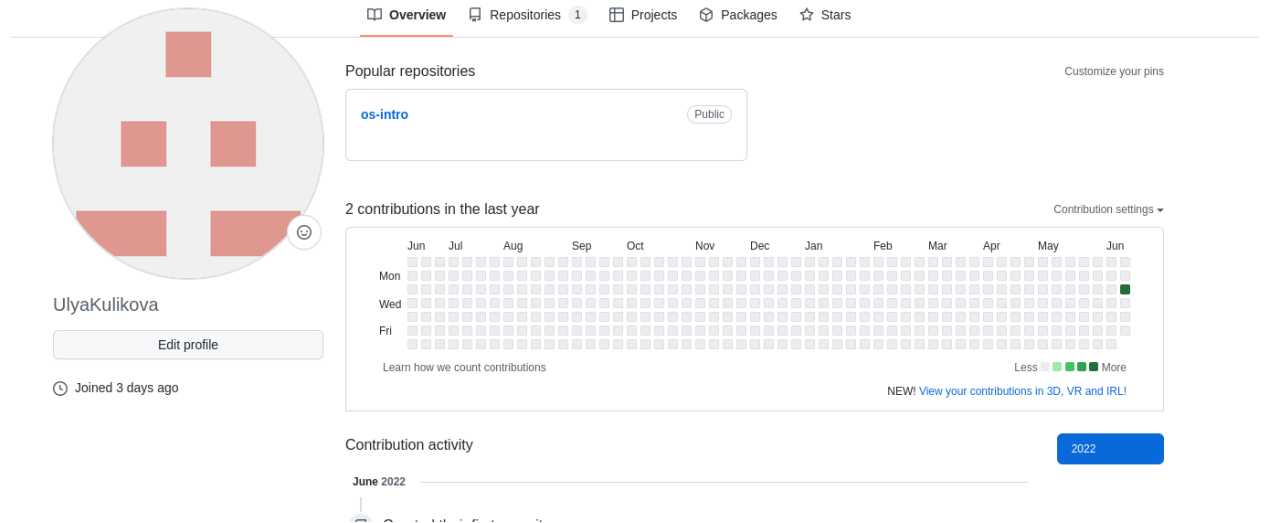
Цель работы

Изучить идеологию и применение средств контроля версий.

Ход работы

Настройка git

1) Создание учётной записи на Github



2) Настройка системы контроля версий git

```
yulia@yulia-VirtualBox:~$ git config --global user.name "Sergey Zaytsev"
yulia@yulia-VirtualBox:~$ git config --global user.email "sergeo.ok@gmail.com"
```

Подключение репозитория к GitHub

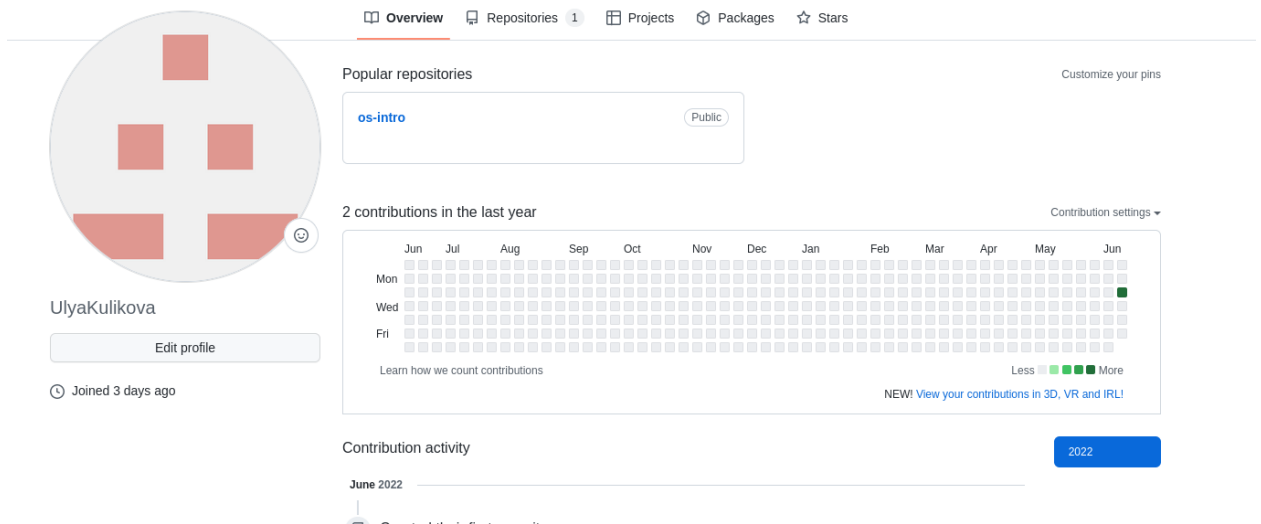
1) генерируем ssh ключ

```
yulia@yulia-VirtualBox:~$ ssh-keygen -C "Yulia Kulikova jili.lina.z@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yulia/.ssh/id_rsa):
Created directory '/home/yulia/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/yulia/.ssh/id_rsa
Your public key has been saved in /home/yulia/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:FLjyNfCgVwcy+8k+XqwnHv7NlCNZjW7I1+lzlX2uMt8 Yulia Kulikova jili.lina.z@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|      ..          |
|      + .         |
|      . =.        |
|     o ..+ o      |
|      + .S.o .   o|
|     o +o = o . .+|
|      +.oB B o   o.|
|     .o+=oB o+  ...|
|      =X*. o  *.o.E|
+---[SHA256]-----+
yulia@yulia-VirtualBox:~$ ss
```

2) Копируем полученный ключ и вставляем в гитхаб

```
yulia@yulia-VirtualBox:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQBzse/+qmmlwmXZvT2j1MuTc5q+G8K8mo80ZT89NiewBMP9hYLEJWyW1wIZ
i42M158BYwwrPEsLirUS2vo104vzU/wuLCPwf0YkN+Bz4d8UsRULfPXc7Qu3dV/s4pJ+Km0HDRvul5rqtDz0addXMY0JfyXBH
oIb+E+Twa2guWd61yDuqhLuppjoku9I12DL8IhwzY9VuHmFXedtq095vSpJga/+Py0wHan8EfvknXjW2T7pEytYNTtvM+QKI
12fig+p2EdPBNwyj9270WoN99HbZXICMB3a8CtJg+h08N4gpT53ZWIr0Xya69TvvXY0y4Dc3fT7HIuRtbo38N59+svZs73beM
```

3) Создаём репозиторий на GitHub



Overview Repositories 1 Projects Packages Stars

Popular repositories

os-intro Public

2 contributions in the last year

Contribution settings

Learn how we count contributions

NEW! View your contributions in 3D, VR and IRL!

Contribution activity

2022

June 2022

4) инициализируем репозиторий

```
yulia@yulia-VirtualBox:~$ cd laboratory/
yulia@yulia-VirtualBox:~/laboratory$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/yulia/laboratory/.git/
yulia@yulia-VirtualBox:~/laboratory$
```

5) Подключаемся к репозиторию os-intro

```
yulia@yulia-VirtualBox:~/laboratory$ git remote add origin git@github.com:UlyaKulikova/os-intro.git
yulia@yulia-VirtualBox:~/laboratory$ git branch -M main
```

6) Создаём заготовку для файла README.md и отправляем на Github

```
yulia@yulia-VirtualBox:~/laboratory$ echo "# Лабораторные работы" >> README.md
yulia@yulia-VirtualBox:~/laboratory$ ls
README.md
```

```
yulia@yulia-VirtualBox:~/laboratory$ git add README.md
yulia@yulia-VirtualBox:~/laboratory$ git commit -m "first commit"
[main (root-commit) c4e202d] first commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md
yulia@yulia-VirtualBox:~/laboratory$ git push -u origin main
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 252 bytes | 252.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:UlyaKulikova/os-intro.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
yulia@yulia-VirtualBox:~/laboratory$
```

Вывод

В ходе выполнения лабораторной работы №2 были освоены навыки администрирования и взаимодействия с децентрализованной системой и операционной системой git для параллельного контроля поддержки программного кода.

Контрольные вопросы

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

Системы контроля версий (VCS) – это программное обеспечение для облегчения работы с изменяющейся информацией. Предназначаются для работы нескольких человек над одним проектом, совместная работа путем изменения файлов в одном репозитории.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- Хранилище – это общее пространство для хранения файлов
- Commit – это команда для записи индексированных изменений в репозитории
- В истории сохраняются все коммиты, по которым можно отследить автора, сообщение, дату и хэш коммита
- Все файлы кроме .git/ называются рабочей копией, и принадлежат пользователю

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы контроля версий сохраняют проект и его файлы на один общий сервер, а децентрализованные системы контроля версий при

каждом копировании данных удаленного репозитория, происходит полное копирование данных в локальный репозиторий. К примеру, централизованные системы контроля версий – SVN, MS TFS, ClearCase; децентрализованные системы контроля версий – Git, Mercurial, Bazaar.

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

- 1) Создаем репозиторий и именуем его
- 2) Добавляем файлы в репозиторий
- 3) Фиксируем с помощью коммитов
- 4) Изменяем файлы репозитория и фиксируем изменения

5. Опишите порядок работы с общим хранилищем VCS.

- 1) Создаем репозиторий, именуем его или присоединяемся к нему в качестве contributor
- 2) Добавляем файлы в репозиторий
- 3) Фиксируем с помощью коммитов
- 4) Изменяем файлы репозитория и фиксируем изменения
- 5) Ждем проверки коммитов при участии других пользователей в общем репозитории

6. Каковы основные задачи, решаемые инструментальным средством git? Систематизация, параллельность разработки программного обеспечения, единое место для хранения файлов проекта

7. Назовите и дайте краткую характеристику командам git.

- Git init – создание репозитория,
- Git clone – клонирование репозитория,
- Git add – добавление изменений в индекс,
- Git reset – удаление изменений из индекса,
- Git commit – коммиты,
- Git rm – удаление файла.

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

К примеру, я использую локальные репозитории для черновых работ по лабораторным работам, а удаленный репозиторий git для их распространения и оценивания преподавателем.

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

Ветви служат для параллельной разработки программного обеспечения, тестирования, отладки и улучшения

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

Игнорирование можно установить для проекта, компьютера и репозитория, цель игнорирования заключается в том, чтобы не отслеживать файлы служебного типа, например временные файлы сборных утилит для проектов или только те файлы, которые полезны при взаимодействии только с очень ограниченным программным обеспечением