

Отчёт по лабораторной работе 2

Первоначальная настройка git

Цвелев С.А. НПИбд-02-22

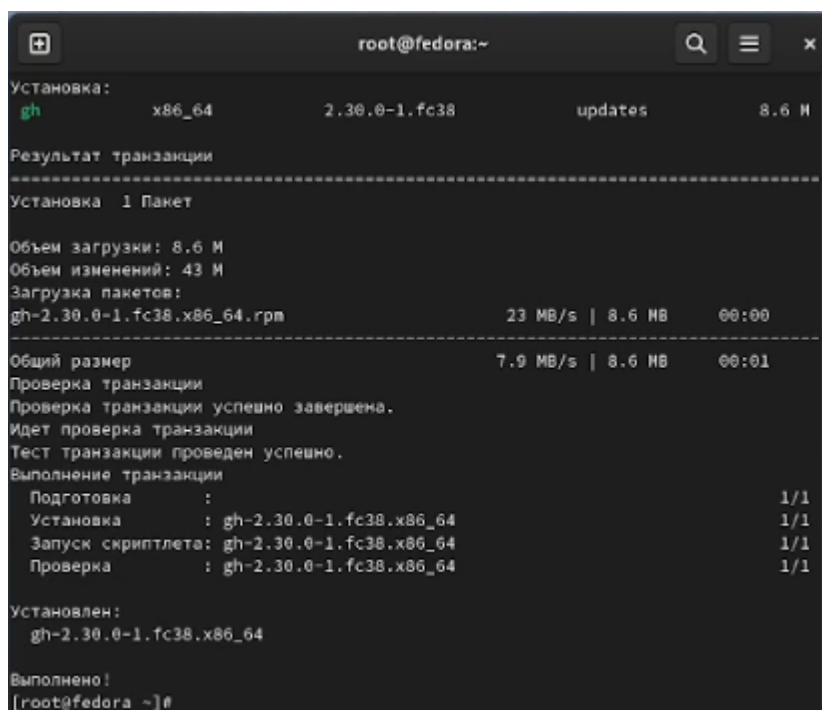
Содержание

1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

2 Выполнение лабораторной работы

Устанавливаем git и gh (рис. 1). Git в моем был уже установлен предварительно.



```
root@fedora: ~  
Установка:  
gh x86_64 2.30.0-1.fc38 updates 8.6 М  
Результат транзакции  
-----  
Установка 1 Пакет  
Объем загрузки: 8.6 М  
Объем изменений: 43 М  
Загрузка пакетов:  
gh-2.30.0-1.fc38.x86_64.rpm 23 MB/s | 8.6 MB 00:00  
-----  
Общий размер 7.9 MB/s | 8.6 MB 00:01  
Проверка транзакции  
Проверка транзакции успешно завершена.  
Идет проверка транзакции  
Тест транзакции проведен успешно.  
Выполнение транзакции  
Подготовка : 1/1  
Установка : gh-2.30.0-1.fc38.x86_64 1/1  
Запуск скриптов: gh-2.30.0-1.fc38.x86_64 1/1  
Проверка : gh-2.30.0-1.fc38.x86_64 1/1  
Установлен:  
gh-2.30.0-1.fc38.x86_64  
Выполнено!  
[root@fedora ~]#
```

Рис. 1. Установка gh

Далее проводим базовую настройку git. На изображении ниже (рис. 2) - задаем имя и email владельца репозитория.



```
[root@fedora ~]# git config --global user.name "sacvelev"  
[root@fedora ~]# git config --global user.email "1132222871@pfur.ru"
```

Рис. 2. Имя и email владельца

Создаём SSH-ключ (рис. 3). Мною была допущена ошибка, когда я проводил эту операцию на root-пользователе. После окончания работы ошибка была исправлена.

```
generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa): rsa  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in rsa  
Your public key has been saved in rsa.pub  
The key fingerprint is:  
SHA256:8j855V326onUvLz/H0J177V4xxjRI0qZ2dpH5y0v5p4 root@fedora  
The key's randomart image is:  
-----[RSA 4096]-----  
|  
|= .  
| = +.xo|  
|. S . + X.x|  
| o o+.x=|  
|. +o.xOx|  
|. + .x=Ox|  
|.o+EooX|  
+----[SHA256]-----
```

Рис. 3. Создание SSH-ключа

Затем, создаём PGP-ключ (рис. 4).

```

Ваше полное имя: sacselev
Адрес электронной почты: 1132222871@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"sacselev <1132222871@pfur.ru>"

сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/53C4996A8EFA8E24F818E9BEF2E2327A756B068E.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2023-06-25 [SC]
    53C4996A8EFA8E24F818E9BEF2E2327A756B068E
uid
    sacselev <1132222871@pfur.ru>
sub   rsa4096 2023-06-25 [E]

```


Рис. 4. Создание PGP-ключа

Затем мы добавляем PGP-ключ в GitHub (рис. 5).

GPG keys

[New GPG key](#)

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



GPG
Email address: 1132222871@pfur.ru
Key ID: F2E2327A756B06BE
Subkeys: 343EDAA796E5CE99
Added on Jun 25, 2023

[Delete](#)

Learn how to [generate a GPG key and add it to your account](#).

Рис. 5. Уже добавленный PGP-ключ

Далее мы настраиваем gh. Выбираем нужные нам параметры. Авторизовываемся с помощью браузера (рис. 6).

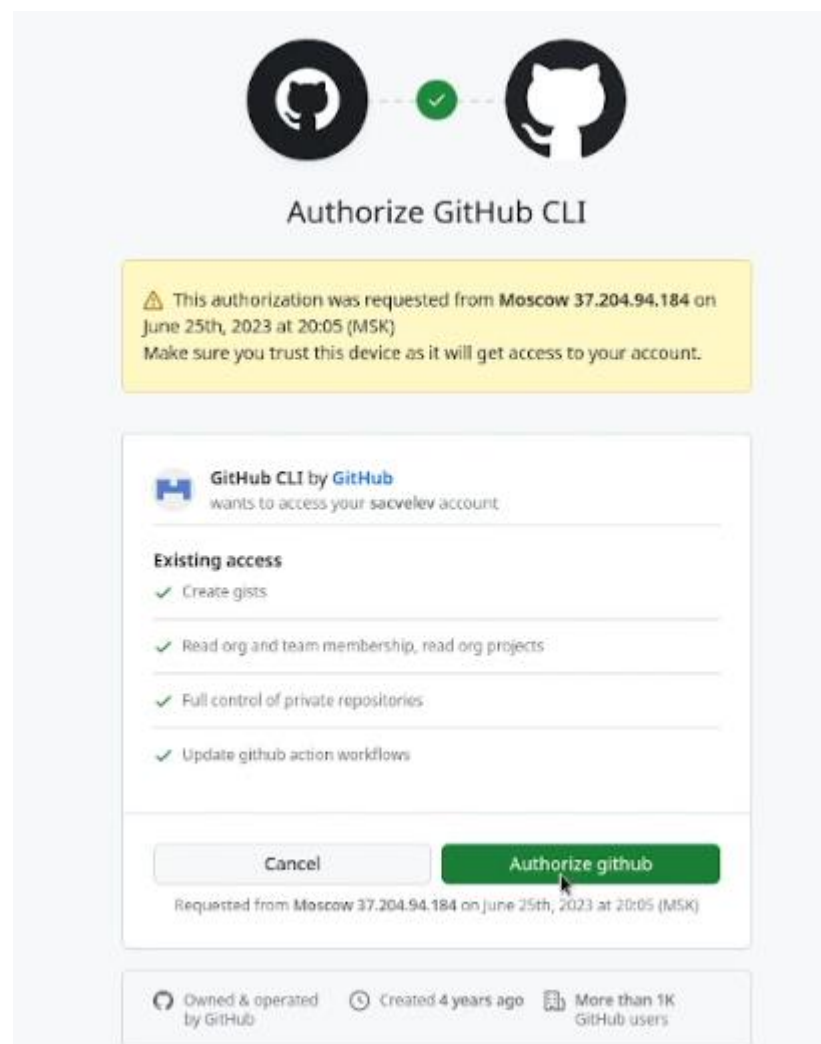


Рис. 6. Авторизация gh через браузер

Далее мы создадим репозиторий курса на основе шаблона (рис. 7). Этот репозиторий у меня уже был, но я создал ещё один для лабораторной работы.

```
[sacvelev@fedora ~]$ mkdir -p ~/work/study/2022-2023/OC/
[sacvelev@fedora ~]$ cd ~/work/study/2022-2023/OC/
[sacvelev@fedora OC]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository sacvelev/study_2022-2023_os-intro on GitHub
```

Рис. 7. Создание репозитория на основе шаблона

После этого мы копируем этот репозиторий на устройство (рис. 8). Из-за вышеупомянутой ошибки я не мог скопировать его и смог только после завершения лабораторной работы. Альтернативным решением было продолжить это на моем Windows, а не виртуальной Fedora.

```
PS C:\Users\Sergey\work\study\2022-2023\oc> git clone --recursive git@github.com:sacvelev/study_2022-2023_os-intro.git
Cloning into 'os-intro'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
Receiving objects: 37% (10/27), 16.93 KiB | 5.64 MiB/s, done.
Receiving objects: 100% (27/27), 16.93 KiB | 5.64 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered
for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path
'template/report'
Cloning into 'C:\Users\Sergey\work\study\2022-2023\OC\os-intro\template\presentation'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Receiving objects: 100% (82/82), 92.90 KiB | 1.90 MiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into 'C:\Users\Sergey\work\study\2022-2023\OC\os-intro\template\report'...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Receiving objects: 100% (101/101), 327.25 KiB | 2.26 MiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 8. Копирование репозитория на устройство

Переходим в каталог курса, удаляем лишний файл package.json и создаем необходимые каталоги. После этого мы отправляем файлы на сервер (рис. 9).

```
PS C:\Users\Sergey\work\study\2022-2023\oc\os-intro> git add .
PS C:\Users\Sergey\work\study\2022-2023\oc\os-intro> git commit -am 'Feat(main): make course structure'
[master 0fb86d0] feat(main): make course structure
2 files changed, 14 deletions(-)
delete mode 100644 package.json
PS C:\Users\Sergey\work\study\2022-2023\oc\os-intro> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:sacvelev/study_2022-2023_os-intro.git
8a72d09..0fb86d0 master -> master
```

Рис. 9. Отправка файлов на сервер

2.2 Ответы на контрольные вопросы

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

Программное обеспечение для работы с постоянно изменяющейся информацией. Позволяет хранить несколько версий одного и того же документа, с возможностью возвращаться к ранним его версиям.

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

Хранилище - место хранения всех версий. Commit - новая версия. История - история изменений. Рабочая копия - текущая версия.

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

Централизованные VCS имеют единое хранилище для всех версий (CVS, Subversion). Децентрализованные имеют каждое хранилище на каждую версию (Git, Mercurial).

Опишите действия с VCS при единоличной работе с хранилищем.

Пользователь размещает новую версию, а может и вернуться к прошлым.

Опишите порядок работы с общим хранилищем VCS.

Общее хранилище работает с несколькими пользователями.

Каковы основные задачи, решаемые инструментальным средством git?

Git помогает работать над кодом в команде и хранит информацию об изменениях.

Назовите и дайте краткую характеристику командам git.

add - добавляет содержимое в индекс, rm - удаляет содержимое из индекса, commit - сохраняет изменения, mv - перемещение, clean - очищение, push - отправка на сервер

Приведите примеры использования при работе с локальным и удалённым репозиториями.

Отчёт по лабораторной отправляется сначала в локальный репозиторий, а после отправляется в удалённый

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

Ветки разделяют проект на несколько частей, чтобы несколько пользователей работали над проектом и не влезали в ветки друг друга

Как и зачем можно игнорировать некоторые файлы при commit?

commit нужно делать не со всеми файлами, а только с теми, которые были изменены или добавлены в ходе работы.

3 Вывод

Я приобрёл практические навыки установки и настройки ОС на виртуальную машину.