

Лабораторная работа №2 Первоначальная настройка git

Дисциплина Операционные системы

Рулев Иван Николаевич НПМбв-02-20"

Содержание"

Цель работы	1
Выполнение лабораторной работы	1
Настройка GIT	1
Создание рабочего пространства	4
Ответы на контрольные вопросы	4
Выводы.....	5

Цель работы

- Изучить применение средств контроля версий
- Освоить умения по работе с git'ом

Выполнение лабораторной работы

Настройка GIT

Сконфигурируем git и создадим SSH ключ

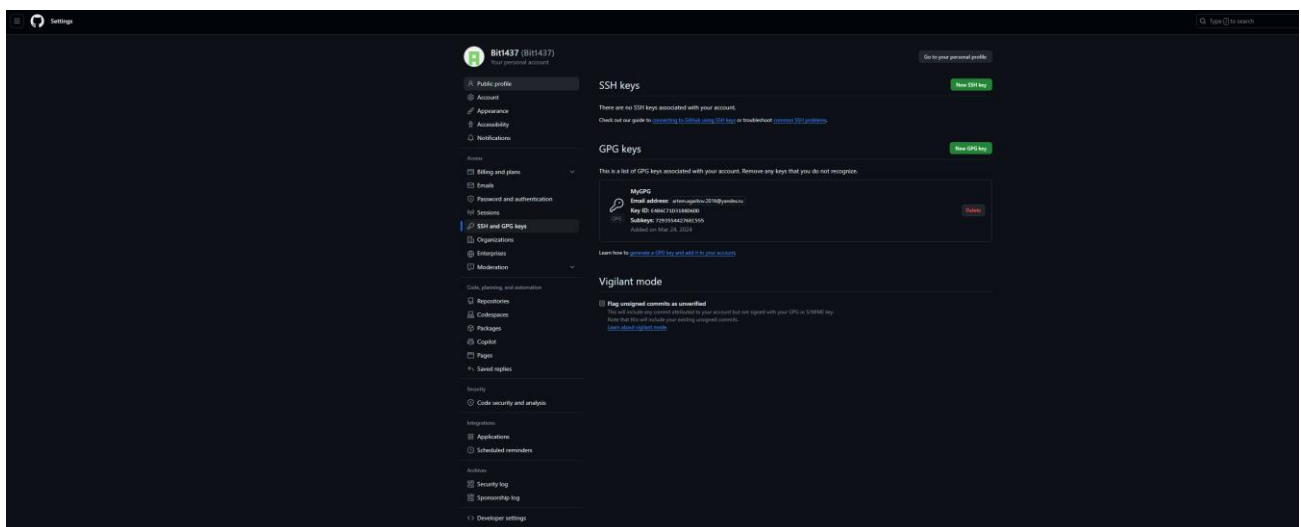
```
root@fedora:~  
становлен:  
gh-2.45.0-1.fc39.x86_64  
  
Выполнено!  
root@fedora:~# git config --global user.name "AAgarkov"  
root@fedora:~# git config --global user.email "artem.agarkov.2016@yandex.ru"  
root@fedora:~# git config --global core.quotepath false  
root@fedora:~# git config --global init.defaultBranch master  
root@fedora:~# git config --global core.autocrlf input  
root@fedora:~# git config --global core.safecrlf warn  
root@fedora:~# ssh-keygen -t rsa -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_rsa  
Your public key has been saved in /root/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:puKfLnfMjCZUuCX6BWey2+mPqUMtVebBelGvyb7XvGA root@fedora  
The key's randomart image is:  
----[RSA 4096]-----+  
+  
o.* |  
o +*.. |  
Boo. |  
o+=oS |  
.O*==.* |  
..+oB.Eo= |  
o.+..O..o |  
oB+. .. |  
-----[SHA256]-----+  
root@fedora:~# ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/root/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_ed25519
```

```
root@fedora:~# ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/root/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_ed25519  
Your public key has been saved in /root/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:fIl+CbiIpEAnHG5PWG5Ca0CH8mhyI0tggHFU2UEIHhU root@fedora  
The key's randomart image is:  
+--[ED25519 256]--+  
|B***E*o. |  
|O=B.o . |  
|+&.= |  
|Bo@ o . . |  
|=+.o . S o |  
|oo . . o o . |  
|. . . . . o |  
| . |  
| |  
+-----[SHA256]-----+  
root@fedora:~#
```

Создадим PGP ключ

```
root@fedora:~  
  
0 = не ограничен  
<n> = срок действия ключа - n дней  
<n>w = срок действия ключа - n недель  
<n>m = срок действия ключа - n месяцев  
<n>y = срок действия ключа - n лет  
Срок действия ключа? (0) 0  
Срок действия ключа не ограничен  
Все верно? (y/N) y  
  
GnuPG должен составить идентификатор пользователя для идентификации ключа.  
  
Ваше полное имя: Artem  
Адрес электронной почты: artem.agarkov.2016@yandex.ru  
Примечание:  
Вы выбрали следующий идентификатор пользователя:  
"Artem <artem.agarkov.2016@yandex.ru>"  
  
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O  
Необходимо получить много случайных чисел. Желательно, чтобы Вы  
в процессе генерации выполняли какие-то другие действия (печать  
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 (печать  
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/EC098C64619744E255B9C911E486C71D3188D6DD.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub  rsa4096 2024-03-24 [SC]  
      EC098C64619744E255B9C911E486C71D3188D6DD  
uid                Artem <artem.agarkov.2016@yandex.ru>  
sub  rsa4096 2024-03-24 [E]  
  
root@fedora:~#
```

С помощью команды `gpg --armor --export <PGP Fingerprint> | xclip -sel clip` скопируем PGP ключ.
И вставим в наш Github.



Авторизируемся в Github с помощью команды `gh`

```
/ Authentication complete.  
- gh config set -h github.com git_protocol https  
/ Configured git protocol  
/ Logged in as Bit1437  
aagarkov@fedora:~/work/study/2022-2023/Операционные системы$ S
```

Создание рабочего пространства

Склонируем репозиторий на локальную машину

```
aagarkov@fedora:~/work/study/2022-2023/Операционные системы$ gh repo create study_2022-2023_os-intro --template=yamadharm/course-directory-student-template --public  
✓ Created repository Bit1437/study_2022-2023_os-intro on GitHub  
https://github.com/Bit1437/study_2022-2023_os-intro
```

Ответы на контрольные вопросы

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

Системы контроля версий (VCS) предназначены для отслеживания изменений в программном коде и обеспечения коллективной разработки.

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

Хранилище: Место, где хранятся все изменения и версии программного кода.

Commit: Отдельное изменение или набор изменений в коде, зафиксированное в системе контроля версий.

История: Последовательность коммитов, отображающая эволюцию кода.

Рабочая копия: Локальная копия проекта, с которой работает разработчик.

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

Децентрализованные VCS копируют всю историю изменений на каждый клиентский компьютер, в то время как централизованные VCS хранят все изменения на центральном сервере и клиенты получают только последние версии файлов. Примеры децентрализованных VCS: Git, Mercurial. Примеры централизованных VCS: Subversion, CVS.

Опишите действия с VCS при единоличной работе с хранилищем.

При индивидуальной разработке пользователь клонирует проект на свой компьютер, вносит изменения и создает новые версии, коммитя их в системе контроля версий.

Опишите порядок работы с общим хранилищем VCS.

Пользователь получает версию проекта из центрального хранилища, вносит изменения, коммитит их и отправляет обратно в хранилище.

Каковы основные задачи, решаемые инструментальным средством git?

Git используется для разработки проектов в команде, контроля изменений в файлах и возможности сохранения нескольких состояний проекта.

Назовите и дайте краткую характеристику командам git.

`git add` - добавляет изменения для коммита.

`git commit` - сохраняет изменения в репозитории с названием.

`git push` - отправляет изменения на удаленный репозиторий.

`git config` - позволяет изменить настройки Git.

Приведите примеры использования при работе с локальным и удалённым репозиториями.

В локальном репозитории разработчик может вносить изменения в код и коммитить их без доступа к сети. В удаленном репозитории команда разработчиков может совместно работать над проектом, обмениваясь изменениями через централизованный сервер.

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

Ветви используются для параллельной разработки функций или исправлений, чтобы избежать конфликтов между изменениями и обеспечить безопасное тестирование нового кода.

Как и зачем можно игнорировать некоторые файлы при commit?

Файлы могут быть проигнорированы с помощью файла `.gitignore`, чтобы избежать загрязнения репозитория лишними или конфиденциальными файлами.

Выводы

Мы изучили идеологию применения средств контроля версий и освоили базовые команды git'a.