

Лабораторная работа №2

Настройка git

Жибицкая Евгения Дмитриевна

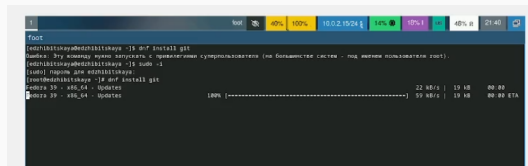
Российский университет дружбы народов, Москва, Россия

Цель

Изучение работы и назначения системы контроля версий git, приобретение навыков по работе с ней.

Выполнение лабораторной работы

Установка необходимых программ



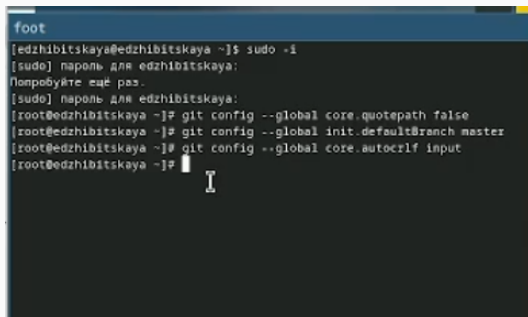
```
1
root 40% 100% 10.0.2.15/24 14% 10% 48% 21.40
root
edzhilitskaya@edzhilitskaya:~$ sudo apt install git
Building dependency tree.
Reading state information.
git is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not installed.
edzhilitskaya@edzhilitskaya:~$ sudo -i
[sudo] password for edzhilitskaya:
edzhilitskaya@edzhilitskaya:~$ sudo apt install git
edzhilitskaya@edzhilitskaya:~$
edzhilitskaya@edzhilitskaya:~$
```

Рис. 1: Установка git

```
fatal: не удалось автоматически определить адрес электронной почты (попытка: fedzhibitskaya@edzhibitskaya.(none))
fedzhibitskaya@edzhibitskaya os-intro]$ git config --global user.email "evgeniazhbitskaya.com"
fedzhibitskaya@edzhibitskaya os-intro]$ git config --global user.name Evgenia Zhibitskaya
>
```

Рис. 2: Ввод почты и имени пользователя

Базовая настройка. Установка параметров




```
foot
[edzhibitskaya@edzhibitskaya ~]$ sudo -i
[sudo] пароль для edzhibitskaya:
Попробуйте ещё раз.
[sudo] пароль для edzhibitskaya:
[root@edzhibitskaya ~]# git config --global core.quotePath false
[root@edzhibitskaya ~]# git config --global init.defaultBranch master
[root@edzhibitskaya ~]# git config --global core.autocrlf input
[root@edzhibitskaya ~]#
```

Рис. 3: Настройка параметров

Создание ключей и привязка к github

```
[edzhibitskaya@edzhibitskaya ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: создан каванор "/home/edzhibitskaya/.gnupg"
gpg: /home/edzhibitskaya/.gnupg/trustdb.gpg: создана таблица доверия
[edzhibitskaya@edzhibitskaya ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/edzhibitskaya/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/edzhibitskaya/.ssh/id_rsa
Your public key has been saved in /home/edzhibitskaya/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:mHt5Z79aLY3hT1u0g5yC/1XaNOx1M3wLvz/qj5Sg51Y edzhibitskaya@edzhibitskaya
The key's randomart image is:
+--[RSA 4096]-----
  o . . .
  o S = . . *
  . o.Oo o.O
  .E.o.B =o M
  .+..+ ..+e+
  .o. .o**o
+----[SHA256]-----
[edzhibitskaya@edzhibitskaya ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/edzhibitskaya/.ssh/id_ed25519):
/home/edzhibitskaya/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/edzhibitskaya/.ssh/id_ed25519
Your public key has been saved in /home/edzhibitskaya/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:T55aq73aA0z0Fu058sy5ned7w5BgM6cvVWw2V9f0e/N edzhibitskaya@edzhibitskaya
The key's randomart image is:
+--[ED25519 256]---
  . . . .
  . . . .
  .B . o .
  oo.. .
  =So. . oo.
  Bx.o o+
  .+o E]
```

Рис. 4: Ключ ssh и ключ pgp


SSH

Sway

SHA256:mHt5Z79aLY3hT1u0g5yC/1XaNOx1M3wLvz/qj5Sg51Y

Added on Feb 22, 2024 by GitHub CLI

Last used within the last week — Read/write


Delete

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

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

pgp

Email address: evgeniazhibitskaya@gmail.com

Key ID: 0FCC68F3A9048888

Subkeys: FC43757F68B28562

Added on Feb 22, 2024

Delete

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

Рис. 5: Добавление ключей по их отпечатку на github


```
edzhibitskaya@edzhibitskaya ~$ git config --global user.signingkey evgeniazhbitskaya@gmail.com
edzhibitskaya@edzhibitskaya ~$ git config --global commit.gpgsign true
edzhibitskaya@edzhibitskaya ~$ git config --global gpg.program $(which gpg2)
edzhibitskaya@edzhibitskaya ~$ cd /path/to/repo
```

Рис. 6: Добавление автоматической подписи коммитов

Создание и клонирование шаблона

```
[edzhbitskaya@edzhbitskaya ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/edzhbitskaya/.ssh/id_rsa.pub
? Title for your SSH key? key
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: 5161.6881
Press Enter to open github.com in your browser...
Authentication complete.
gh config set -h github.com git_protocol ssh
Configured git protocol
Uploaded the SSH key to your GitHub account: /home/edzhbitskaya/.ssh/id_rsa.pub
Logged in as jamezhbits
You were already logged in to this account
[edzhbitskaya@edzhbitskaya ~]$ cd -/work/study/2022-2023/"Операционные системы"
bash: cd: /home/edzhbitskaya/work/study/2022-2023/Операционные системы: Пер: ramora @jamezhbits или karamora
[edzhbitskaya@edzhbitskaya ~]$ cd -/work/study/2023-2024/Операционные системы/
[edzhbitskaya@edzhbitskaya Операционные системы]$ git clone --recursive git@github.com:owner/study_2022-2023_os-intro.git os-intro
bash: owner: Пер: ramora @jamezhbits или karamora
[edzhbitskaya@edzhbitskaya Операционные системы]$ git clone --recursive git@github.com:jamezhbits/study_2023-2024_os-intro.git os-intro
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
```

Рис. 7: Работа с шаблоном

```
foot
Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
[root@edzhbitskaya ~]# cd /opt/edzhbitskaya
[edzhbitskaya@edzhbitskaya ~]$ echo os-intro > COURSE
[edzhbitskaya@edzhbitskaya ~]$ make
Usage:
  make <target>

Targets:
  list              List of courses
  prepare           Generate directories structure
  submodule         Update submodules

[edzhbitskaya@edzhbitskaya ~]$ make list
net-admin          Администрирование локальных сетей
net-os-admin       Администрирование сетевых подсистем
arch-pc            Архитектура ЭВМ
sciprog-intro      Введение в научное программирование
infosec           Информационная безопасность
computer-practice Компьютерный практикум по статистическому анализу д
онных
mathsec           Математические основы защиты информации и информации
онной безопасности
mathmod           Математическое моделирование
simulation-networks Моделирование сетей передачи данных
sciprog           Научное программирование
os-intro           Операционные системы

[edzhbitskaya@edzhbitskaya ~]$ make prepare
[edzhbitskaya@edzhbitskaya ~]$ make submodule
git submodule update --init --recursive
git submodule foreach 'git fetch origin; git checkout $(git rev-parse --
abbrev-ref HEAD); git reset --hard origin/$(git rev-parse --abbrev-ref H
EAD); git submodule update --recursive; git clean -dfx'
Entering 'template/presentation'
Указатель HEAD сейчас на коммите 40a1761 Merge branch 'release/1.0.3'
Entering 'template/report'
Указатель HEAD сейчас на коммите 7c31ab8 Merge branch 'release/1.0.4'
[edzhbitskaya@edzhbitskaya ~]$ ls
CHANGELOG.md  labs      prepare  README.en.md  template
config        LICENSE  presentation  README.git-flow.md
COURSE       Makefile  project-personal  README.md
[edzhbitskaya@edzhbitskaya ~]$ git add .
[edzhbitskaya@edzhbitskaya ~]$ git commit -am 'feat(template)'
```

```
knos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
[edzhbitskaya@edzhbitskaya os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При скатии изменений используется до 2 потоков
Сжатие объектов: 100% (38/38), готово.
Запись объектов: 100% (38/38), 342.11 КиБ | 2.74 МБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно
использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:Janezhbit/study_2023-2024_os-intro.git
 f029ada..1549115 master -> master
[edzhbitskaya@edzhbitskaya os-intro]$
```

Рис. 9: Отправка изменений на github

Контрольные вопросы

Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?
 - Они применяются при работе нескольких человек с одним проектом. При внесении изменений позволяют фиксировать, совмещать и возвращать изменения разных людей
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
 - Хранилище - место, где находятся данные(файлы, коды и тд)
 - Commit - команда, для сохранения изменений
 - История - информация о предыдущих изменениях
 - Рабочая копия - одна из версий проекта, с которой ведется работа(= текущая/основная)

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
 - Централизованные системы предполагают наличие единого репозитория для хранения данных(CVC, Subversion)
 - В децентрализованных системах центральный репозиторий не обязателен(Git, Bazaar)
4. Опишите действия с VCS при единоличной работе с хранилищем. Работа происходит на своем компьютере, сначала обновляются данные, в конце они размещаются в центральном репозитории

5. Опишите порядок работы с общим хранилищем VCS. Для идентификации на сервере необходимы ключи и затем создание репозитория, только затем можно работать на локальной машине. Также в конце изменения добавляются на сервер
6. Каковы основные задачи, решаемые инструментальным средством git?
 - хранение информации о всех изменениях
 - обеспечение удобства командной работы

Ответы на контрольные вопросы

7. Назовите и дайте краткую характеристику командам 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`

7.

- 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 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 checkout -b имя_ветки`

Ответы на контрольные вопросы

8. Приведите примеры использования при работе с локальным и удалённым репозиториями. локальный репозиторий - работа со своими файлами удаленный репозиторий - совместная работа, общий проект
9. Что такое и зачем могут быть нужны ветви (branches)? Ветви это пути к отдельным состояниям (отделы проекта),они дают возможность вносить изменения только в часть проекта и не трогать все вышестоящее
10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорировать файлы можно с помощью .gitignore. Нужно это, например, при наличии ненужных(лишних или созданных автоматически) файлов.

Вывод

В ходе работы была освоена работа с системой контроля версий, был установлен git, проведена авторизация, заданы базовые настройки, создались ключи, клонировался репозиторий и так далее.