

Лабораторная работа №2

Презентация

Андрюшин Н. С.

25 февраля 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Андрюшин Никита Сергеевич
- Студент
- Российский университет дружбы народов
- 1132231848@pfur.ru

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

Создать базовую конфигурацию для работы с git. Создать ключ SSH. Создать ключ PGP. Настроить подписи git. Зарегистрироваться на Github. Создать локальный каталог для выполнения заданий по предмету.

Установка git

Для начала установим git. В моём случае он уже установлен

```
[nsandryushin@nsandryushin ~]$ dnf install git
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на
  большинстве систем - под именем пользователя root).
[nsandryushin@nsandryushin ~]$ sudo dnf install git
[sudo] пароль для nsandryushin:
Последняя проверка окончания срока действия метаданных: 2:01:32 назад, П
т 23 фев 2024 19:21:01.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 1: Установка git

Теперь установим gh

```
[nsandryushin@nsandryushin ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 2:01:46 назад, П
т 23 фев 2024 19:21:01.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh         x86_64       2.43.1-1.fc39  updates      9.1 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 9.1 М
Объем изменений: 46 М
Продолжить? [д/н]: y
Загрузка пакетов:
gh-2.43.1-1.fc39.x86_64.rpm          327 kB/s | 9.1 MB  00:28
-----
Общий размер          315 kB/s | 9.1 MB  00:29
-----
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
  Подготовка      :
  Установка       : gh-2.43.1-1.fc39.x86_64      1/1
  Запуск скрипта  : gh-2.43.1-1.fc39.x86_64      1/1
  Проверка        : gh-2.43.1-1.fc39.x86_64      1/1
Установлен:
gh-2.43.1-1.fc39.x86_64
Выполнено!
```

Рис. 2: Установка gh

Далее, зададим имя для владельца репозитория. В данном случае это моё имя

```
[nsandryushin@nsandryushin ~]$ git config --global user.name "Nikita Andryushin"
```

Рис. 3: Указание имени

Теперь зададим почту. Я задал почту, на которую у меня зарегистрирован аккаунт на github

```
[nsandryushin@nsandryushin os-intro]$ git config --global user.email "mega_nikitos111@mail.ru"
```

Рис. 4: Указание почты

Настроим кодировку utf8 в выводе сообщений git

```
[nsandryushin@nsandryushin ~]$ git config --global core.quotePath false
```

Рис. 5: Настройка кодировки utf8

Базовая настройка git

Зададим имя начальной ветки, настроим параметры autocrlf и safecrlf

```
[nsandryushin@nsandryushin ~]$ git config --global init.defaultBranch master  
[nsandryushin@nsandryushin ~]$ git config --global core.autocrlf input  
[nsandryushin@nsandryushin ~]$ git config --global core.safecrlf warn
```

Рис. 6: Настройка git

Создание ключа ssh

Создадим ключ RSA размером 4096 бит

```
[nsandryushin@nsandryushin ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nsandryushin/.ssh/id_rsa):
Created directory '/home/nsandryushin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nsandryushin/.ssh/id_rsa
Your public key has been saved in /home/nsandryushin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RyDm2EEeT0qlby0UPGcGc13TL0sb+Gh1ox4hr+LJ5cc nsandryushin@nsandryushin
The key's randomart image is:
+---[RSA 4096]----+
|      .0+B... .oo |
|      * 0o=+ . .o |
|      , = 0= .   o |
|      o o   .   |
|      S o. +   |
|      . o. + =.. |
|      . E *.+. |
|      o.+ +.+ |
|      .=.o.. |
+---[SHA256]-----+
```

Рис. 7: Создание ключа RSA

Создание ключа ssh

Теперь создадим ключ по алгоритму ed25519

```
[nsandryushin@nsandryushin ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/nsandryushin/.ssh/id_ed25519)
:
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nsandryushin/.ssh/id_ed25519
Your public key has been saved in /home/nsandryushin/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:JBI3EaMOR5Bt+qDzrx1Y0eS+oHSDriu02ch+bsADvQU nsandryushin@nsandryushin
The key's randomart image is:
+--[ED25519 256]--+
|  .+o Bo          |
|  E.oB o          |
|  ..+= + .        |
|  . +=. + o        |
| o.o=* . S         |
| o*.=.o .         |
| .o* . .          |
| +=. + .          |
| X+*+o            |
+----[SHA256]-----+
```

Рис. 8: Создание ключа ed25519

Создание ключа pgp

Теперь создадим ключ gpg. Выбираем из предложенных вариантов первый тип (RSA and RSA), размер ключа задаём 4096 бит и делаем срок действия ключа неограниченным

```
[nsandryushin@nsandryushin ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.3; Copyright (C) 2023 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/nsandryushin/.gnupg'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
  (10) ECC (только для подписи)
  (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  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
```

Рис. 9: оздание ключа pgp (1)

Создание ключа pgp

После нас попросят ввести свои данные. Мы вводим имя и адрес электронной почты. После этого соглашаемся с генерацией ключа

```
GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: Nikita Andryushin
Адрес электронной почты: mega_nikitos111@mail.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"Nikita Andryushin <mega_nikitos111@mail.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 (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: /home/nsandryushin/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/nsandryushin/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/nsandryushin/.gnupg/openpgp-revo
cs.d/8C5B506CED0386A3BFFC80D950243248E62B33F0.rev'.
Открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2024-02-23 [SC]
      8C5B506CED0386A3BFFC80D950243248E62B33F0
uid           Nikita Andryushin <mega_nikitos111@mail.ru>
sub   rsa4096 2024-02-23 [E]
```

Рис. 10: оздание ключа pgp (2)

Добавление PGP ключа в GitHub

Далее, выводим список pgp ключей

```
[nsandryushin@nsandryushin ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n
, 0m, 0f, 1u
[keyboxd]
-----
sec   rsa4096/50243248E62B33F0 2024-02-23 [SC]
      8C5B506CEDD386A3BFFC80D950243248E62B33F0
uid           [ абсолютно ] Nikita Andryushin <mega_nikitos111@mail
.ru>
ssb   rsa4096/7E067DD165DB2558 2024-02-23 [E]
```

Рис. 11: Список pgp ключей

Копируем наш ключ в буфер обмена

```
[nsandryushin@nsandryushin ~]$ gpg --armor --export mega_nikitos111@mail.ru | xclip -sel clip
```

Рис. 12: Копирование ключа

Добавление PGP ключа в GitHub

Вставляем этот ключ на гитхаб, и задаём ему имя. Я выбрал имя Sway

Add new GPG key

Title

Sway

Key

```
vv
gv12xclvwzgiZMbsDMAw7j17VlesK3zeVAKpLO8DhLLEKJETG5tl+Hd
oFNHEajL
cUtkmAmMx1tp0WhIqE2qUYarvPfmDTIC0rxI9wztezuokJDRhmM03xFc
2vrt/d1me
y56uwEMxb7g914WVcQSR7xBSYMz6jWICVws3Ku6UYLg6loBNR3Klo
pOg425I3ARt
2udMdEg72a1t4A==
=ZuG3
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис. 13: Вставка ключа в GitHub

Настройка автоматических подписей коммитов git

Теперь производим настройку автоматических подписей

```
[nsandryushin@nsandryushin ~]$ git config --global user.signingkey mega_nikitos111@mail.ru  
[nsandryushin@nsandryushin ~]$ git config --global commit.gpgsign true  
[nsandryushin@nsandryushin ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 14: Настройка автоматических подписей коммитов git

Настройка gh

После, нам нужно авторизоваться в github с помощью gh. Мы выбираем сайт для авторизации (GitHub.com), после выбираем предпочитаемый протокол (SSH), публичный SSH ключ (id_rsa.pub), и имя для ключа (Sway). В качестве способа авторизации выбираем авторизацию через браузер

```
[nsandryushin@nsandryushin ~]$ 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/nsandryushin/.ssh/id_rsa.pub
? Title for your SSH key: Sway
? How would you like to authenticate GitHub CLI? Login with a web browser
?

! First copy your one-time code: 1DE0-199A
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/nsandryushin/.ssh/id_rsa.pub
✓ Logged in as DrNikiyProgrammingAccount
```

Рис. 15: Авторизация в gh

Теперь создаём рабочую директорию курса и переходим в неё

```
[nsandryushin@nsandryushin ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"  
[nsandryushin@nsandryushin ~]$ cd ~/work/study/2023-2024/"Операционные системы"
```

Рис. 16: Создание рабочей директории и переход в неё

Создание репозитория курса на основе шаблона

Далее, создаём репозиторий для лабораторных работ из шаблона

```
[nsandryushin@nsandryushin Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository DrNikiyProgrammingAccount/study_2023-2024_os-intro on GitHub
https://github.com/DrNikiyProgrammingAccount/study_2023-2024_os-intro
```

Рис. 17: Создание репозитория курса

Создание репозитория курса на основе шаблона

И клонируем его к себе на компьютер

```
[nsandryushin@nsandryushin Операционные системы]$ git clone --recursive  
git@github.com:DrNikiyProgrammingAccount/study_2023-2024_os-intro.git os  
-intro  
Клонирование в «os-intro»...
```

Рис. 18: Клонирование репозитория

Настройка каталога курса

Переходим в него с помощью `cd` и удаляем ненужные файлы (`package.json`) и создаём необходимые каталоги, записав в файл `COURSE` строку `os-intro` (это наш текущий курс) и прописываем `make prepare` для того, чтобы нужные нам каталоги создались

```
[nsandryushin@nsandryushin Операционные системы]$ cd ~/work/study/2023-2024/"Операционные системы"/os-intro
[nsandryushin@nsandryushin os-intro]$ rm package.json
[nsandryushin@nsandryushin os-intro]$ echo os-intro > COURSE
[nsandryushin@nsandryushin os-intro]$ make
Usage:
  make <target>

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

[nsandryushin@nsandryushin os-intro]$ make prepare
```

Рис. 19: Удаление ненужных файлов и использование `make`

Теперь добавляем нашу папку для отправки

```
[nsandryushin@nsandryushin os-intro]$ git add .
```

Рис. 20: Использование git add

Настройка каталога курса

Делаем коммит, в котором указываем, что мы сделали структуру курса

```
[nsandryushin@nsandryushin os-intro]$ git commit -am 'feat(main): make course structure'
[master aa04ea9] feat(main): make course structure
361 files changed, 98413 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numer
ic.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__
.py
```

Рис. 21: Использование git commit

Настройка каталога курса

И отправляем файлы на сервер GitHub с помощью команды push

```
[nsandryushin@nsandryushin os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.12 КиБ | 1.41 МиБ/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:DrNikiyProgrammingAccount/study_2023-2024_os-intro.git
  733e347..aa04ea9  master -> master
```

Рис. 22: Использование git push

Была произведена установка git, проведена его первоначальная настройка, были созданы ключи для авторизации и подписи, а также создан репозиторий курса из предложенного шаблона