

Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютеров и операционных систем

Румянцев Артём Олегович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
3.1	Установка программного обеспечения	8
3.2	Базовая настройка git	9
3.3	Создание ключа SSH	9
3.4	Создание Ключа GPG	10
3.5	Регистрация на GitHub	11
3.6	Добавление ключа GPG в GitHub	12
3.7	Настроить подписи Git	13
3.8	Настройка gh	13
3.9	Создание репозитория курса на основе шаблона	14
3.10	Выводы	16

Список иллюстраций

3.1	Рис 1	8
3.2	Рис 1	8
3.3	Рис 2	9
3.4	Рис 3	9
3.5	Рис 4	9
3.6	Рис 5	9
3.7	Рис 6	10
3.8	Рис 7	10
3.9	Рис 8	11
3.10	Рис 9	11
3.11	Рис 10	12
3.12	Рис 11	12
3.13	Рис 12	13
3.14	Рис 13	13
3.15	Рис 14	13
3.16	Рис 15	14
3.17	Рис 16	14
3.18	Рис 17	15
3.19	Рис 18	15
3.20	Рис 19	15
3.21	Рис 20	15
3.22	Рис 21	16

Список таблиц

1 Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

2 Задание

1.Создать базовую конфигурацию для работы с git 2.Создать ключ SSH 3.Создать ключ GPG 4.Настроить подписи Git 5.Зарегистрироваться на GitHub 6.Создать локальный каталог для выполнения заданий по предмету # Теоретическое введение

Основные команды git

Перечислим наиболее часто используемые команды 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`

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 add имена_файлов`

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

`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 commit` новой ветки, базирующейся на текущей:

`git checkout -b имя_ветки`

переключение на некоторую ветку:

`git checkout имя_ветки`

(при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий:

`git push origin имя_ветки`

слияние ветки с текущим деревом:

`git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки:

`git branch -d имя_ветки`

принудительное удаление локальной ветки:

`git branch -D имя_ветки`

удаление ветки с центрального репозитория:

`git push origin :имя_ветки`

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Установка необходимого программного обеспечения git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh` (рис.1).

```
root@10:~# dnf install git
Fedora 39 - x86_64 - Updates          95 kB/s | 19 kB      00:00
Fedora 39 - x86_64 - Updates          2.3 MB/s | 3.2 MB    00:01
Последняя проверка окончания срока действия метаданных: 0:00:07 назад, Сб 02 мар
2024 09:50:33.
Пакет git-2.44.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 3.1: Рис 1

```
root@10:~# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:41 назад, Сб 02 мар
2024 09:50:33.
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]: д
Загрузка пакетов:
```

Рис. 3.2: Рис 1

3.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту (рис 2)

```
root@10:~# git config --global user.name "Artem Romyancev"
root@10:~# git config --global user.email "1132231426@rudn.ru"
```

Рис. 3.3: Рис 2

Настраиваю utf-8 в выводе сообщений git для их корректного отображения

```
root@10:~# git config --global core.quotePath false
```

Рис. 3.4: Рис 3

Начальной ветке задаю имя master

```
root@10:~# git config --global init.defaultBranch master
```

Рис. 3.5: Рис 4

Задаю параметры autocrlf и safecrlf для корректного отображения конца строки

```
root@10:~# git config --global core.autocrlf input
root@10:~# git config --global core.safecrlf warn
```

Рис. 3.6: Рис 5

3.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa

```

root@10:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key
Your public key has been saved in key.pub
The key fingerprint is:
SHA256:WG9zzR4UcfIZiG42wM6NIq4Vqe60P6rCFY92Qg8KSGM root@10.0.2.15
The key's randomart image is:
+---[RSA 4096]-----+
|      .      =o.|
| E      o . . =o|
|+ . . . + = . . .|
|o + + + = * + |
| . o 0 + S * o + |
| . * * . o . . |
| . + = . |
|...+
|+o=+.
+---[SHA256]-----+

```

Рис. 3.7: Рис 6

Создаю ключ ssh по алгоритму ed 25519

```

root@10:~# 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:XE3ptioLw5izGiitoT8AGgDln9y0Fxgo2G+zN9EWJUY root@10.0.2.15
The key's randomart image is:
+--[ED25519 256]--+
|o+. .o+E.. .. |
|o.o..o.o.o o. |
|. .o+ o..... |
|o .+..oo. o |
|o. . o oS . . |
|oo .+o . |
|+.o +.+ . |
|oo.. o o. . |
|o.oo. .o |
+---[SHA256]-----+
root@10:~#

```

Рис. 3.8: Рис 7

3.4 Создание Ключа GPG

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю макс. длину ключа; 4096, оставляю неограниченный срок действия ключа. Далее отвечаю на вопросы программы о личной информации

```

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

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: RumyantsevArtem
Адрес электронной почты: 1132231426@rudn.ru
Примечание:

```

Рис. 3.9: Рис 8

3.5 Регистрация на GitHub

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполнил и проводил его настройку, поэтому просто вхожу в свой аккаунт

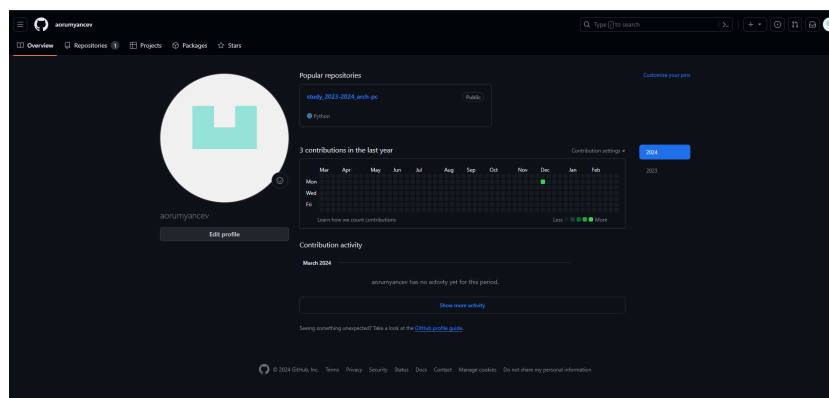


Рис. 3.10: Рис 9

3.6 Добавление ключа GPG в GitHub

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа (последовательность байтов для идентификации более длинного, по сравнению с самим отпечатком, ключа), он стоит после знака слеша, копирую его в буфер обмена

```
root@i0:~# 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
[keyboard]
-----
sec   rsa4096/258ECFD6A5F8CC24 2024-03-02 [SC]
      5F105B484AD968D2324C4881258ECFD6A5F8CC24
uid   [ абсолютно ] RumyancevArtem <1132231426@rudn.ru>
ssb   rsa4096/FE62F0356EAC7088 2024-03-02 [E]
root@i0:~#
```

Рис. 3.11: Рис 10

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip.

```
root@i0:~# gpg --armor --export 258ECFD6A5F8CC24 | xclip -sel clip
```


Рис. 3.12: Рис 11

Открываю настройки GitHub, ищу среди них добавление GPG ключа. Нажимаю на New GPG key и вставляю в поле ключ из буфера обмена. Я добавил ключ GPG на GitHub

PGP keys

[New GPG key](#)

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



Email address: 1132231426@rudn.ru
Key ID: 258ECFD6A5F8CC24
Subkeys: FE62F0356EAC7088
Added on Mar 2, 2024

[Delete](#)

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

Рис. 3.13: Рис 12

3.7 Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов.

```
root@10:~# git config --global user.signingkey 258ECFD6A5F8CC24
root@10:~# git config --global commit.gpgsign true
root@10:~# git config --global gpg.program $(which gpg2)
```

Рис. 3.14: Рис 13

3.8 Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер

```
root@10:~# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser
```

Рис. 3.15: Рис 14

Завершаю авторизацию на сайте

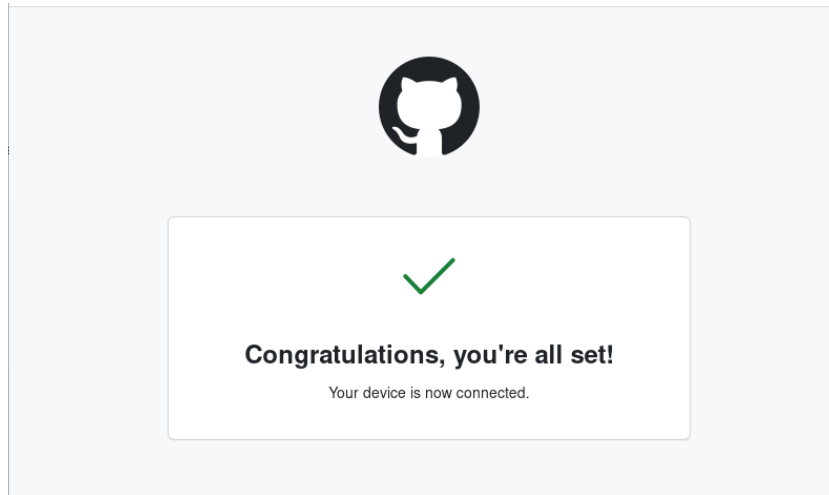


Рис. 3.16: Рис 15

Вижу сообщение о завершении авторизации под именем aorumyancev

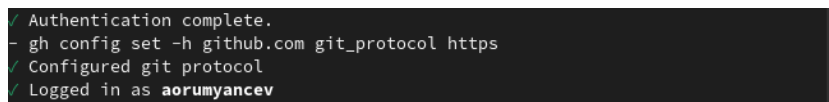


Рис. 3.17: Рис 16

3.9 Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты `mkdir` и флага `-p`, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты `cd` перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду `gh repo create study_2023-2024_os-intro-template yamadharma/course-derctory-student-trmplate-public`, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом `https`, а не `ssh`, потому что при авторизации в `gh` выбрала протокол `https`

```

aorumyancev@10: ~/work/study/2023-2024/Операционные системы$ git clone --recursive https://github.com/aorumyancev/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 414.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/aorumyancev/work/study/2023-2024/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.

```

Рис. 3.18: Рис 17

Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls`

```

aorumyancev@10: ~/work/study/2023-2024/Операционные системы$ cd os-intro
aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$

```

Рис. 3.19: Рис 18

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile`

```

aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$ rm package.json
aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$ echo os-intro > COURSE
aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$ make

```

Рис. 3.20: Рис 19

Добавляю все новые файлы для отправки на сервер(сохраняю добавленные изменения с помощью команды `git add` и комментирую их с помощью `git commit`

```

aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$ git add .
aorumyancev@10: ~/work/study/2023-2024/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 68f63e4] feat(main): make course structure

```

Рис. 3.21: Рис 20

Отправляю файлы на сервер с помощью `git push`.

```
aorumyancev@10:~/work/study/2023-2024/Операционные системы/os-intro$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 281 байт | 281.00 КиБ/с, готово.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aorumyancev/study_2023-2024_os-intro.git
   6ad0403..68f63e4  master -> master
aorumyancev@10:~/work/study/2023-2024/Операционные системы/os-intro$
```

Рис. 3.22: Рис 21

3.10 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, освоил умение по работе с git.