

РУДН. Операционные системы

Отчёт по лабораторной работе №2

Косинов Никита Андреевич, НПИМбв-02-20

Содержание

1	Цель работы	5
2	Ход работы	6
3	Настройка github и git	7
4	Создание структурированного пространства для работы	12
5	Контрольные вопросы	15
6	Выводы	16

Список иллюстраций

3.1	Настройка git	7
3.2	Создание SSH-ключа по алгоритму RSA	7
3.3	Создание SSH-ключа по алгоритму ed25519	8
3.4	Запись SSH ключа	8
3.5	Запись SSH ключа	8
3.6	Генерация PGP-ключа	9
3.7	Генерация PGP-ключа	9
3.8	Запись GPG ключа	9
3.9	Запись GPG ключа	10
3.10	Режим бдительности	10
3.11	Финальная настройка	10
3.12	Финальная настройка	11
3.13	Финальная настройка	11
4.1	Создание папки	12
4.2	Копирование шаблона	13
4.3	Репозиторий yamadharma	13
4.4	Внесение изменений	14

Список таблиц

1 Цель работы

При работе большой команды людей над некоторым проектом неизменно возникают ситуации с накопленными одновременно изменениями. Для решения этой проблемы разработаны различные системы контроля версий проектов, чтобы любой член команды вовремя получал изменения проекта и мог их учитывать.

Цель данной работы - приобретение теоретических и практических навыков по работе с системой контроля версий на примере **Git**.

2 Ход работы

Лабораторная работа выполнена в терминале **ОС Linux** и хостинге хранения проектов **Github** с использованием **VCS Git**. Действия по лабораторной работе представлены в следующем порядке: 1. Настройка **git** и учётной записи **github**; 2. Создание рабочего пространства; 3. Самостоятельная работа.

3 Настройка github и git

Настраиваем рабочий репозиторий и предварительную конфигурацию Git.

1. У нас уже есть учётная запись на сайте **github**, так что сразу переходим к настройке репозитория и связи его с глобальным. Указываем имя пользователя и свой **e-mail**. Настраиваем также **utf-8** и верификацию.

```
nakosinov@dk3n62 ~ $ git --version
git version 2.43.0
nakosinov@dk3n62 ~ $ git config --global user.name "Loin Cosinus"
nakosinov@dk3n62 ~ $ git config --global user.name "kosinovna-1"
nakosinov@dk3n62 ~ $ git config --global user.name "kosinov.n@gmail.com"
nakosinov@dk3n62 ~ $ git config --global core.quotepath false
nakosinov@dk3n62 ~ $ git config --global init.defaultBranch master
nakosinov@dk3n62 ~ $ git config --global core.autocrlf input
nakosinov@dk3n62 ~ $ git config --global core.safecrlf warn
nakosinov@dk3n62 ~ $
```

Рис. 3.1: Настройка git

2. Создаём пару **SSH**-ключей: публичный и приватный. Публичный - чтобы другие узлы видели нашу работу, приватный - чтобы определить узел, из которого мы будем работать.

```
nakosinov@dk3n62 ~ $ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.ssh/id_rsa): git_key_rsa.txt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in git_key_rsa.txt
Your public key has been saved in git_key_rsa.txt.pub
The key fingerprint is:
SHA256:C/1BBgM20Si6QwBqLvikwKncNS5et1DV3xycU1kfZdw nakosinov@dk3n62
The key's randomart image is:
+---[RSA 4096]-----+
|o  ==o      +O|
|o  ....o    ooE|
|oo   o. . . +.|
|*..  +.  . + .|
|==. oo.S    o |
|B+. o oo o    |
|oo.o + .o     |
| . o o .      |
| . .         |
+---[SHA256]-----+
```

Рис. 3.2: Создание SSH-ключа по алгоритму RSA

```
nakosinov@dk3n62 ~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.ssh/id_ed25519): git_key_ed.t
xt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in git_key_ed.txt
Your public key has been saved in git_key_ed.txt.pub
The key fingerprint is:
SHA256:3aoC+VTacUwPFCdk2FhvPHxYgbg+DXEeoEuxNsuoyY nakosinov@dk3n62
The key's randomart image is:
+--[ED25519 256]--+
|      . oB=o      |
|    oo.+& o      |
| ..+.*o% o      |
| o.*oOo* .      |
| . *S+.+. .      |
| o = o .        |
| = o .          |
| E o o .        |
| o . .          |
+---[SHA256]-----+
nakosinov@dk3n62 ~$
```

Рис. 3.3: Создание SSH-ключа по алгоритму ed25519

3. Привязываем созданный по алгоритму **ed25519** ключ к своему аккаунту на **GitHub**

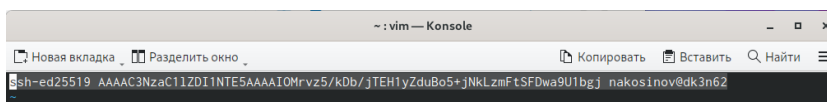


Рис. 3.4: Запись SSH ключа

Add new SSH Key

Title
OS-RUDN

Key type
Authentication Key

Key
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOMrvz5/kDb/jTEH1yZduBo5+jNkLzmFtSFDwa9U1bgj nakosinov@dk3n62

Add SSH key

Рис. 3.5: Запись SSH ключа

3. Генерируем **GPG**-ключ для корректной работы коммитов. Указываем необходимую конфигурацию, своё полное имя и почту. Дополнительный пароль для защиты не ставим.


```

nakosinov@dk3n62 ~ $ gpg --full-generate-key
gpg (GnuPG) 2.2.42; 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.

Выберите тип ключа:
  (1) RSA и RSA (по умолчанию)
  (2) DSA и Elgamal
  (3) DSA (только для подписи)
  (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
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: Nikita Kosinov
Адрес электронной почты: kosinov.n@gmail.com

```

Рис. 3.6: Генерация PGP-ключа

```

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Nikita Kosinov
Адрес электронной почты: kosinov.n@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Nikita Kosinov <kosinov.n@gmail.com>"

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: создан каталог '/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.gnupg/openpgp-revocs.d/B26C1D627
3E9E36235ED44EA50D4E35623C659EB.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2024-03-19 [SC]
     B26C1D6273E9E36235ED44EA50D4E35623C659EB
uid                               Nikita Kosinov <kosinov.n@gmail.com>
sub  rsa4096 2024-03-19 [E]

nakosinov@dk3n62 ~ $

```

Рис. 3.7: Генерация PGP-ключа

4. Выводим отпечаток ключа. По нему копируем сам ключ для дальнейшей привязке к своему аккаунту на **GitHub**.

```

nakosinov@dk3n62 ~ $ 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
/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.gnupg/pubring.kbx
-----
sec  rsa4096/50D4E35623C659EB 2024-03-19 [SC]
     B26C1D6273E9E36235ED44EA50D4E35623C659EB
uid                               [ абсолютно ] Nikita Kosinov <kosinov.n@gmail.com>
ssb  rsa4096/F2D6B2556FAB175 2024-03-19 [E]

nakosinov@dk3n62 ~ $ gpg --armor --export 50D4E35623C659EB | xclip -sel clip
nakosinov@dk3n62 ~ $

```

Рис. 3.8: Запись GPG ключа

Add new GPG key

Title

OS-RUDN

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGX5ycoBEADUpyxsgR+mmfZ67p/tsdC1/qJvQr3h09hnON/dyXWC6E613CZr
FBekhb1h4tqB956sXavojRqYO8i6RH7X1Q3XK2OtgLv6U9u6SXDoNANyHVwazs
cmc8laGVK4nOem9hjshqMxNwCmXvTI26UjKwVKGe24T/ylmVUjhZpNslzYMHUM
qM1e4IMEcFMt9UPLS0P7USOknP/187vDUjTSFK4o0u6xQwrp6/C5Kz+IMlmrl2/v
si6DvhE+nQAh5AFD8Tc/u5fW88UwGmPyNLcmJuWn45Mu5128Y2wP+jL3MA/gHzS
GgpOo2wMSZwU70ZTOz+P32YBSAHetQetHlUjLHocmmQovwqo56kU7TonZ9dk7mV
aUMM1PH5krUyVCpgsJtqWWDG+DVdApeRbC3xcW8oCuR9XoOE1cyntCstMmsfX7pkh

-----
```

Add GPG key

Рис. 3.9: Запись GPG ключа

5. Включаем режим бдительности для отслеживания неподписанных ком- МИНТОВ

Vigilant mode

☒ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
[Learn about vigilant mode.](#)

Рис. 3.10: Режим бдительности

6. Настраиваем автоматические подписи коммитов и настраиваем **gh**.

```
nakosinov@dk3n62 ~ $ vim git_key_ed.txt.pub
nakosinov@dk3n62 ~ $ vim git_key_ed.txt.pub
nakosinov@dk3n62 ~ $ gpg --armor --export 50D4E35623C659EB | xclip -sel clip
nakosinov@dk3n62 ~ $ git config --global user.signingkey 50D4E35623C659EB
nakosinov@dk3n62 ~ $ git config --global commit.gpgsign true
nakosinov@dk3n62 ~ $ git config --global gpg.program $(which gpg2)
nakosinov@dk3n62 ~ $ 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? [Use arrows to move, type to filter]
  /afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/.ssh/id_rsa.pub
> Skip
```

Рис. 3.11: Финальная настройка

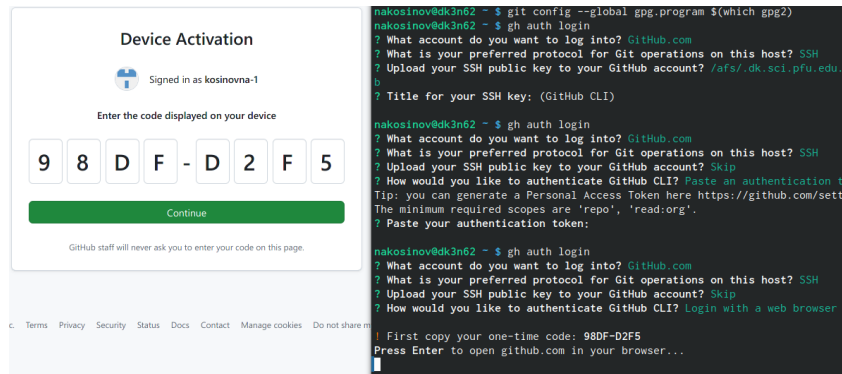


Рис. 3.12: Финальная настройка

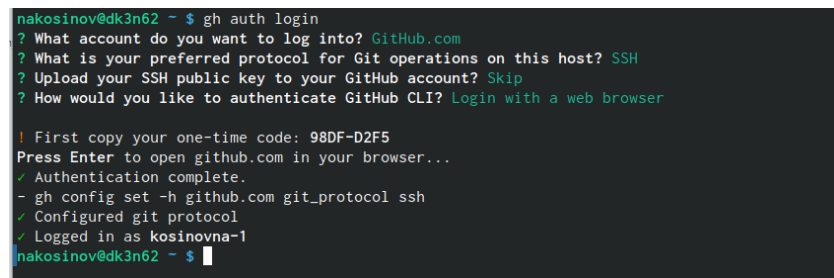


Рис. 3.13: Финальная настройка

4 Создание структурированного пространства для работы

С помощью терминала и интерфейса **github** создаём удобную для чтения и работы файловую систему будущих лабораторных работ.

1. Создаём папку для предмета “Операционные системы” командой **mkdir** с использованием ключа **p** для одновременного создания вложенных папок. Переходим в созданную папку и копируем шаблон репозитория пользователя **yamadharma**.

```
nakosinov@dk3n62 ~$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
nakosinov@dk3n62 ~$ cd ~/work/study/2022-2023/"Операционные системы"
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository kosinovna-1/study_2022-2023_os-intro on GitHub
```

Рис. 4.1: Создание папки

```

nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы $ git clone --recursive git@github.com:kosinovna-1/study_2022-2023_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 | 3.72 МБ/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) зарегистри-
рован по пути «template/report»
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/work/study/2022-2023/Операционные системы/os-intro/
template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 1.14 МБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/work/study/2022-2023/Операционные системы/os-intro/
template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.80 КиБ | 2.19 МБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67caeb8a19ef8028ced88e'
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы $

```

Рис. 4.2: Копирование шаблона

2. Проверяем, что всё заработало успешно

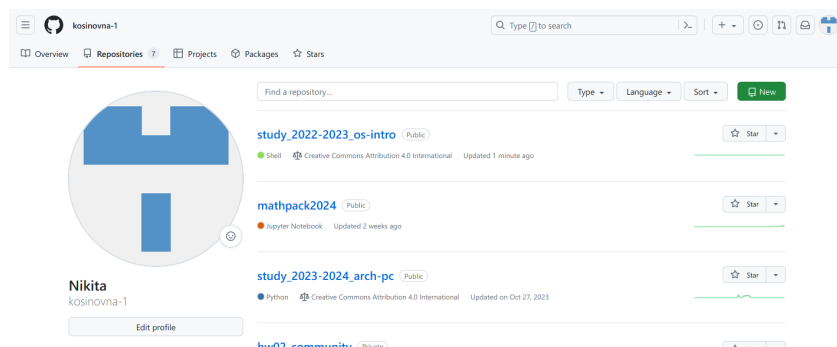


Рис. 4.3: Репозиторий yamadharma

3. Удаляем лишние папки из шаблона, создаём необходимые, а также зали- ваем изменения на репозиторий, сделав коммит **make course structure**.

```

nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы $ cd os-intro
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $ rm package.json
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $ echo os-intro > COURSE
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $ make
Usage:
  make <target>

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

nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $ git add .
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -am 'feat(main): make course structure'
[master 8b79b0a] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
nakosinov@dk3n62 ~/work/study/2022-2023/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), 946 байтов | 946.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:kosinovna-1/study_2022-2023_os-intro.git
 c0c7612..8b79b0a master -> master
nakosinov@dk3n62 ~/work/study/2022-2023/Операционные системы/os-intro $

```

Рис. 4.4: Внесение изменений

4. После завершения отчёта планируем также загрузить его на репозиторий

5 Контрольные вопросы

1. Система контроля версий (**Version Control System, VCS**) - это программное обеспечение для отслеживания изменений в документах, что обеспечивает удобную работу с нескольких устройств и, возможно, несколькими разработчиками. **VCS** также позволяет фиксировать версии, делать откаты на более ранние, разрешать конфликты изменений, следить, кто именно внёс и какие исправления и т.п.
2. Хранилище - место, где располагается проект. Commit - это способ сохранять изменения. История в Git - это место хранения всех сделанных commit-ов. Рабочая копия - это “снимок” одной из версий проекта, с которым можно работать на локальном компьютере.
3. Централизованная VCS - это клиент-серверское приложение, когда проект существует в единственном экземпляре и хранится в общедоступном месте - сервере. Примеры: CVS, SVN. Децентрализованная (распределённая) VCS - приложение, позволяющее хранить копию проекта у каждого разработчика, работающего с ним. При этом есть место хранения всех изменений проекта. Примеры: Mercurial, Git.

6 Выводы

В данной лабораторной работе мы узнали о способе бесконфликтного одновременного изменения проекта с использованием средств контроля версий на примере **VCS Git**. Поняли, как связывать глобальный репозиторий с локальными его копиями, с каждой из которых работает отдельный разработчик, загружать изменения и указывать на них.

[1].

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.