

Первоначальна настройка git

Лабораторная работа №2

Казначеев Сергей Ильич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	17

Список иллюстраций

3.1	sc1	7
3.2	screen2	7
3.3	sc3	8
3.4	sc_3	8
3.5	sc_3_1	8
3.6	sc4	8
3.7	sc5	9
3.8	sc6	10
3.9	sc7	11
3.10	sc8	11
3.11	sc9	12
3.12	sc10	12
3.13	sc11	12
3.14	sc13	13
3.15	sc14	13
3.16	sc15	14
3.17	sc16	14
3.18	sc17	15
3.19	sc18	15
3.20	sc19	16
3.21	sc20	16

Список таблиц

1 Цель работы

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

2 Задание

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.

3 Выполнение лабораторной работы

Для начала установим Git

```
[iskaznacheev@iskaznacheev ~]$ sudo dnf install git
[sudo] пароль для iskaznacheev:
Обновление и загрузка репозитория:
  Fedora 41 - x86_64 - Updates      100% | 29.8 KiB/s | 17.7 KiB | 00m
  Fedora 41 - x86_64 - Updates      100% | 2.1 MiB/s | 3.8 MiB | 00m
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
Нечего делать.
```

Рис. 3.1: sc1

Теперь установим gh

```
[iskaznacheev@iskaznacheev ~]$ sudo dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет      Арх.  Версия      Репозиторий  Размер
Установка:
  gh       x86_64  2.65.0-1.fc41  updates      42.6 MiB

Сводка транзакции:
  Установка:      1 пакета

Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.
После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB,
удаление 0 B).
Is this ok [y/N]:
```

Рис. 3.2: screen2

Далее задаем имя владельца репозитория

```
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "Kava-45"  
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "mechtitana@gmail.com"  
"  
[iskaznacheev@iskaznacheev ~]$ git config --global core.quotepath false  
[iskaznacheev@iskaznacheev ~]$
```

Рис. 3.3: sc3

Далее задаем почту владельца git

```
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "Kava-45"  
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "mechtitana@gmail.com"  
"  
[iskaznacheev@iskaznacheev ~]$ git config --global core.quotepath false  
[iskaznacheev@iskaznacheev ~]$
```

Рис. 3.4: sc_3

Далее настроим кодировку UTF-8 в выводе сообщения git

```
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "Kava-45"  
[iskaznacheev@iskaznacheev ~]$ git config --global user.name "mechtitana@gmail.com"  
"  
[iskaznacheev@iskaznacheev ~]$ git config --global core.quotepath false  
[iskaznacheev@iskaznacheev ~]$
```

Рис. 3.5: sc_3_1

Зададим имя начальной ветки настроим параметры autocrlf и safecrlf

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

Рис. 3.6: sc4

Создадим ключ RSA размером 4096 бит


```

[iskaznacheev@iskaznacheev ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/iskaznacheev/.ssh/id_rsa):
Created directory '/home/iskaznacheev/.ssh'.
Enter passphrase for "/home/iskaznacheev/.ssh/id_rsa" (empty for no passphrase)
Enter same passphrase again:
Your identification has been saved in /home/iskaznacheev/.ssh/id_rsa
Your public key has been saved in /home/iskaznacheev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5hPO+8Y32+SQWVh@xyEJhd0yMoxdMUjskYqfPDTivAk iskaznacheev@iskaznacheev
The key's randomart image is:
+---[RSA 4096]-----+
|      o..o=ooo+|
|      B + +..o|
|    . B . + . |
|    o = * . o |
|    o = S o . . |
|  E o X . + |
|    . o +. + . |
|    o  oo o= |
|    .o...oo |
+-----[SHA256]-----+

```

Рис. 3.7: sc5

Теперь создаем ключ по алгоритму ed22519

```

[iskaznacheev@iskaznacheev ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/iskaznacheev/.ssh/id_ed25519):
Enter passphrase for "/home/iskaznacheev/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iskaznacheev/.ssh/id_ed25519
Your public key has been saved in /home/iskaznacheev/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:pj7eHInj89JSCTBRMwquH7DPpNzBzMT8Rf0kk0lGAcc iskaznacheev@iskaznacheev
The key's randomart image is:
+--[ED25519 256]--+
|      .o=++      |
|      o. +E      |
|      oo + o     |
|      = .+ . =    |
|      . = .S..   |
|      O . =oo     |
|      o B =oo     |
|      . B ==+..   |
|      o =.+*+     |
+-----[SHA256]-----+
[iskaznacheev@iskaznacheev ~]$ 

```

Рис. 3.8: sc6

Теперь создаем gpg, выбираем из предложенных вариантов первый тип(тип RSA and RSA),размер 4096 бит и делаем срок ключа неограниченным

```

[iskaznacheev@iskaznacheev ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 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/iskaznacheev/.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

GnuPG должен составить идентификатор пользователя для идентификации ключа.

```

Рис. 3.9: sc7

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

```

Ваше полное имя: Sergey Kaznacheev
Адрес электронной почты: mechtitana@gmail.com

```

Рис. 3.10: sc8

Далее выводим список ключей gpg

```
[iskaznacheev@iskaznacheev ~]$ 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,
lu
[keyboard]
-----
sec   rsa4096/7AF44B29F2494555 2025-02-24 [SC]
      ACD948A102E4848580AF8BE87AF44B29F2494555
uid           [ абсолот ] Sergey Kaznacheev <nechtitana@gmail.com>
ssb   rsa4096/7F62A14C0FEEA451 2025-02-24 [E]
```

Рис. 3.11: sc9

Копируем наш ключ в буфер обмена

```
[iskaznacheev@iskaznacheev ~]$ gpg --armor --export nechtitana@gmail.com | xclip -
sel clip
```

Рис. 3.12: sc10

Вставляем ключ на Github и задаем ему имя

Title

Sway

Key

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

mQINBGe8N2oBEAC7qTpeojvUE0HeHmNEUu3MU8B2Q9hvTmkzxnS2
vHeWMH6Q2o4q
GLyIuBxonBGrP5lzUDmfhRmyTQ4011cGP79yuxWNnm7knXWCWnGTE
Q+0RbPtRKsc
SwGg9TKSzxWyujv9R7W3B+mITDz7XFgUL1ONGpSfg2bA8RRrzbrncBh
9HyeqmN/U
TqAts86Mng+P+g+8eO/9FSIP6YPPTm+ds2zv+WK0mhUSIkPSUjh8R1rpj

Add GPG key

Рис. 3.13: sc11

```
[iskaznacheev@iskaznacheev ~]$ git conf
l.com
[iskaznacheev@iskaznacheev ~]$ git conf
[iskaznacheev@iskaznacheev ~]$ git conf
[iskaznacheev@iskaznacheev ~]$
```

Теперь произведем автоматическую настройку подписей

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

```
[iskaznacheev@iskaznacheev ~]$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/iskaznacheev/.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: BF22-7118
Press Enter to open https://github.com/login/device 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/iskaznacheev/.ssh/id_rsa.pub
✓ Logged in as Kava-45
```

Рис. 3.14: sc13

Теперь создаем рабочую директорию курса и переходим в неё

```
[iskaznacheev@iskaznacheev ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[iskaznacheev@iskaznacheev ~]$ cd ~/work/study/2022-2023/"Операционные системы"
```

Рис. 3.15: sc14

Создаем репозиторий для лабораторных работ из шаблона

```
[iskaznacheev@iskaznacheev Операционные системы]$ gh repo create s
tudy_2024-2025_os-intro --template=yamadharma/course-directory-stu
dent-template --public
✓ Created repository Kava-45/study_2024-2025_os-intro on GitHub
https://github.com/Kava-45/study_2024-2025_os-intro
[iskaznacheev@iskaznacheev Операционные системы]$
```

Рис. 3.16: sc15

И клонируем его к себе на компьютер

```
[iskaznacheev@iskaznacheev Операционные системы]$ git clone --recursive git@github.co
m:Kava-45/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvvV6TuJJhbpZisF/zLDA0zPM5vHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
```

Рис. 3.17: sc16

Переходим в него с помощью `cd` и удаляем файл(`package.json`) и создаем необходимые каталоги записав в файл `COURSE` строку `echo os-intro` и прописываем `make prerepare` для того, чтобы нужные нам каталоги создались

```

[iskaznacheev@iskaznacheev Операционные системы]$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro
[iskaznacheev@iskaznacheev os-intro]$ rm package.json
[iskaznacheev@iskaznacheev os-intro]$ echo os-intro > COURSE
[iskaznacheev@iskaznacheev os-intro]$ make
Usage:
  make <target>

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

[iskaznacheev@iskaznacheev os-intro]$ make prepare
[iskaznacheev@iskaznacheev os-intro]$ ll
итого 56
-rw-r--r--. 1 iskaznacheev iskaznacheev 5230 фев 24 12:33 CHANGELOG.md
drwxr-xr-x. 1 iskaznacheev iskaznacheev  24 фев 24 12:33 config
-rw-r--r--. 1 iskaznacheev iskaznacheev   9 фев 24 12:34 COURSE
drwxr-xr-x. 1 iskaznacheev iskaznacheev  192 фев 24 12:34 labs
-rw-r--r--. 1 iskaznacheev iskaznacheev 18657 фев 24 12:33 LICENSE
-rw-r--r--. 1 iskaznacheev iskaznacheev   980 фев 24 12:33 Makefile
-rw-r--r--. 1 iskaznacheev iskaznacheev    0 фев 24 12:34 prepare
drwxr-xr-x. 1 iskaznacheev iskaznacheev   78 фев 24 12:34 presentation
drwxr-xr-x. 1 iskaznacheev iskaznacheev  114 фев 24 12:34 project-personal
-rw-r--r--. 1 iskaznacheev iskaznacheev   152 фев 24 12:33 README.en.md
-rw-r--r--. 1 iskaznacheev iskaznacheev  5653 фев 24 12:33 README.git-flow.md
-rw-r--r--. 1 iskaznacheev iskaznacheev  4384 фев 24 12:33 README.md
drwxr-xr-x. 1 iskaznacheev iskaznacheev   36 фев 24 12:33 template
[iskaznacheev@iskaznacheev os-intro]$

```

Рис. 3.18: sc17

Теперь добавляем нашу папку для отправки

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

Рис. 3.19: sc18

Делаем коммит в котором указываем что мы сделали структуру курса

```

ic.csi
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_secnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/__init__
.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/pandocat
tributes.py

```

Рис. 3.20: sc19

И отправляем файлы на Github с помощью команды push

```

[iskaznacheev@iskaznacheev os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.31 КиБ | 2.46 МБ/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:Kava-45/study_2024-2025_os-intro.git
d8cc9a5..e059854 master -> master

```

Рис. 3.21: sc20

4 Выводы

Я изучил идеологию и применение средств контроля версий и освоил умения по работе с git