

Лабораторная работа №2

Первоначальная настройка git

Полякова Юлия Александровна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	12
4	Выводы	14

Список иллюстраций

2.1	Установка ПО	6
2.2	Базовая настройка	7
2.3	Создание ключей ssh	7
2.4	Создание ключа gpg	8
2.5	Добавление ключа gpg	9
2.6	Настройка подписей коммитов	9
2.7	Клонирование репозитория в папку os-intro	10
2.8	Настройка каталога курса	11
2.9	Отправление файлов на сервер	11

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Установка git и gh командой dnf install (рис. 2.1).

```
[yapolyakoval@yapolyakoval ~]$ sudo -i
[sudo] password for yapolyakoval:
[root@yapolyakoval ~]# dnf install git
Updating and loading repositories:
  Fedora 41 - x86_64 - Updates
  Fedora 41 - x86_64 - Updates
Repositories loaded.
Package "git-2.48.1-1.fc41.x86_64" is already installed.

Nothing to do.
[root@yapolyakoval ~]# dnf install gh
Updating and loading repositories:
Repositories loaded.
Package                               Arch      Version
Installing:
  gh                                     x86_64    2.65.0-1.fc41

Transaction Summary:
  Installing:      1 package

Total size of inbound packages is 10 MiB. Need to download 10 MiB.
After this operation, 43 MiB extra will be used (install 43 MiB, remove 0 B).
Is this ok [y/N]: y
[1/1] gh-0:2.65.0-1.fc41.x86_64
-----
[1/1] Total
Running transaction
[1/3] Verify package files
[2/3] Prepare transaction
[3/3] Installing gh-0:2.65.0-1.fc41.x86_64
Complete!
```

Рис. 2.1: Установка ПО

2. Базовая настройка git. Задаем имя и email, настраиваем utf-8, задаем имя начальной ветки и параметры autocrlf и safecrlf (рис. 2.2)

```
[root@yapolyakoval ~]# git config --global user.name "Julia Polyakova"
[root@yapolyakoval ~]# git config --global user.email "yulya.polyakova.07@mail.ru"
[root@yapolyakoval ~]# git config --global core.quotepath false
[root@yapolyakoval ~]# git config --global init.defaultBranch master
[root@yapolyakoval ~]# git config --global core.autocrlf input
[root@yapolyakoval ~]# git config --global core.safecrlf warn
```

Рис. 2.2: Базовая настройка

3. Создание ключей ssh по алгоритмам rsa и ed25519 (рис. 2.3)

```
[root@yapolyakoval ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/id_rsa
Enter passphrase for "/root/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Czb8pXrNVMxiEhtwKp+9u6yHtLSJtDPiDL3J5N16tM root@yapolyakoval.yapolyakoval.net
The key's randomart image is:
+---[RSA 4096]-----+
| +oo . . 0 |
| . = + . o |
| . + . |
| . 0 . |
| . . X . S o |
| o @ B.o.. |
| + * Xo*+o |
| ,0 -+*+E |
| ooo.. |
+---[SHA256]-----+
[root@yapolyakoval ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /root/.ssh/id_ed25519
Enter passphrase for "/root/.ssh/id_ed25519" (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:WuHTVCruz6ghA1+Xk/Gsy8z4dIAAME1rsnbygLJmW8 root@yapolyakoval.yapolyakoval.net
The key's randomart image is:
+---[ED25519 256]---+
| ,0+0 . . |
| . . . . . |
| . . . . . |
| . 0 . . + |
| o o o S.. |
| ++o. E.o.. |
| .ooo+.** |
| |0.+0.*+0+ |
| .oo. .==+. |
+---[SHA256]-----+
[root@yapolyakoval ~]#
```

Рис. 2.3: Создание ключей ssh

4. Создание ключа gpg с типом RSA and RSA, размером 4096 и сроком действия по умолчанию. Также добавляем личную информацию (имя, адрес почты,

пустой комментарий) (рис. 2.4)

```
[root@yapolyakoval ~]# 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: directory '/root/.gnupg' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: yapolyakoval
Email address: yulya.polyakova.07@mail.ru
Comment:
You selected this USER-ID:
  "yapolyakoval <yulya.polyakova.07@mail.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? █
```

Рис. 2.4: Создание ключа gpg

- Учетная запись в GitHub уже была настроена, поэтому переходим к добавлению ключа. Выводим ключ по его отпечатку и копируем в настройки на сайте (рис. 2.5)

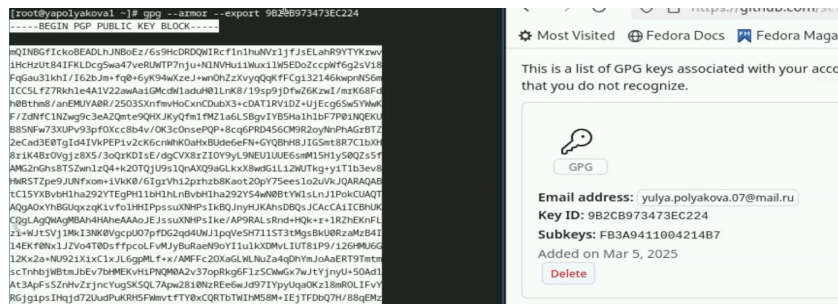


Рис. 2.5: Добавление ключа gpg

6. Настройка автоматических подписей коммитов git (рис. 2.6)

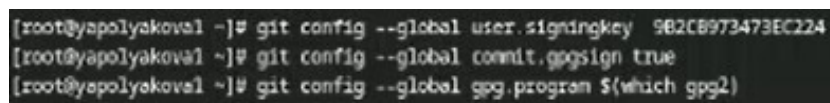


Рис. 2.6: Настройка подписей коммитов

7. Авторизируемся с помощью `gh auth login`. Создаем папку курса по шаблону, переходим в нее. Создаем в терминале репозиторий по указанному шаблону. Далее клонируем получившийся репозиторий в папку `os-intro`. На этом моменте возникла сложность с ключом (`Permission denied (publickey)`), поэтому дополнительно был добавлен на сайт GitHub ключ `ssh`. После этого был успешно клонирован репозиторий (рис. 2.7)

```

yapolyakova1@yapolyakova1:~$ cd ~/work/study/2024-2025/"Операционные системы"
yapolyakova1@yapolyakova1:~/work/study/2024-2025/"Операционные системы"$ git clone --recursive
git@github.com:JuliaMaffin123/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
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)
Receiving objects: 100% (36/36), 19.38 KiB | 19.38 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markd
own-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-templa
te.git) registered for path 'template/report'
Cloning into '/home/yapolyakova1/work/study/2024-2025/Операционные системы/os-intro/template
/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 683.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/yapolyakova1/work/study/2024-2025/Операционные системы/os-intro/template
/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 1.59 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5085c9c72a02bd2fcald4a
6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
yapolyakova1@yapolyakova1:~/work/study/2024-2025/Операционные системы$

```

Рис. 2.7: Клонирование репозитория в папку os-intro

8. Настройка каталога курса (удаление лишних файлов, создание необходи-
мых каталогов командой make) (рис. 2.8)

```

yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json
yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

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

yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ make list
      net-admin   Администрирование локальных сетей
      net-os-admin Администрирование сетевых подсистем
      arch-pc     Архитектура ЗВМ
      sciprog-intro Введение в научное программирование
      netcybersec Защита сетей и кибербезопасность
      simmod      Имитационное моделирование
      infosec     Информационная безопасность
      computer-practice Компьютерная практика по статистическому анализу данных
      mathsec     Математические основы защиты информации и информационной безопасности
      mathmod     Математическое моделирование
      simulation-networks Моделирование сетей передачи данных
      sciprog      Научное программирование
      os-intro     Операционные системы
      os2         Основы администрирования операционных систем
      infosec-intro Основы информационной безопасности

yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ make prepare
yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$ 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 HEAD); git submodule update --recursive; git clean -df'
Entering 'template/presentation'
HEAD is now at c862712 Merge branch 'release/1.0.5'
Entering 'template/report'
HEAD is now at c8622e Merge branch 'release/1.0.5'
yapylykova@yapylykova1:~/work/study/2024-2025/Операционные системы/os-intro$

```

Рис. 2.8: Настройка каталога курса

9. Успешное отправление файлов на сервер командами `git add .` (добавить все файлы), `git commit` (создание коммита) и `git push` (непосредственно отправка файлов на сервер) (рис. 2.9)

```
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_figsnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_secnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxenos/_init_.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxenos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxenos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxenos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
yapolyakova@yapolyakova:~/work/study/2024-2025/0nepauwomee ciktetel/os-intro$ git push
Enumerating objects: 40, done.
Counting objects: 100% (40/40), done.
Delta compression using up to 4 threads
Compressing objects: 100% (30/30), done.
Writing objects: 100% (38/38), 341.66 KiB | 2.85 MiB/s, done.
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: JuliaMaffin123/study_2024-2025_os-intro.git
97b75e9..32d2686 master -> master
yapolyakova@yapolyakova:~/work/study/2024-2025/0nepauwomee ciktetel/os-intro$
```

Рис. 2.9: Отправление файлов на сервер

3 Контрольные вопросы

1. VCS - это инструмент, который помогает разработчикам отслеживать изменения в коде, управлять версиями файлов и координировать работу в команде. Задачи: отслеживание изменений, решение конфликтов, поддержка параллельной разработки, управление версиями, совместная работа и улучшение координации.
2. Хранилище (репозиторий) - место хранения всех версий и служебной информации. Commit - состояние проекта на определенный момент времени или логическая группа изменений, которые пользователь потом отправляет в хранилище. История - сохранение полной истории изменений. Рабочая копия - текущее состояние файлов проекта, полученных из хранилища и возможно измененных.
3. Централизованные имеют одно основное хранилище всего проекта. Каждый пользователь копирует себе изменения оттуда, а потом загружает свои. Пример: Subversion, CVS, TFS, VAULT, AccuRev. Децентрализованные имеют у каждого пользователя свой вариант репозитория, возможно даже не один. Есть возможность добавлять и забирать изменения из любого репозитория. Пример: Git, Mercurial, Bazaar.
4. Получение нужной версии файлов, размещение новой версии в хранилище, обновление рабочей копии, отслеживание и разрешение конфликтов.
5. То же самое, только с более тщательным контролем, “чистыми коммитами”, советуясь с другими пользователями.

6. Отслеживание изменений, добавление новых функций в проект, объединение разных версий кода в один, откат к предыдущей версии, исправление конфликтов.
7. Команды можно разделить на работу с деревом, работу с изменениями, слияние и удаление веток.
8. Например, студент создает удаленный репозиторий, клонирует его себе, пишет программу, делает `commit` и отправляет изменения в локальном репозитории на удаленный. Его друг внес изменения на удаленном репозитории, а студент в следующий раз их получил с помощью `pull`
9. С помощью ветвей можно разрабатывать новый функционал независимо от основного кода. Можно делать откат и переключаться между версиями, не боясь что-то потерять.
10. Игнорировать можно не добавляя их `add`-ом, в случае, если эти файлы не нужны на удаленном репозитории.

4 Выводы

Была изучена идеология и применение средств контроля версий. Были освоены умения по работе с git.