

Лабораторная работа №2

Первоначальная настройка git

Полякова Ю.А.

28 февраля 2007

Российский университет дружбы народов, Москва, Россия

Информация

- Полякова Юлия Александровна
- Студент
- Российский университет дружбы народов
- yulya.polyakova.07@mail.ru
- <https://github.com/JuliaMaffin123>



Вводная часть

- Важно уметь работать со средствами контроля версий
- Это важно для работы над проектом в команде
- Также полезно понимать структуру версий, ветвей и уметь делать “чистые коммиты”

- Средство контроля версий git

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

- Сайт GitHub
- VCS git
- Ключи ssh и gpg
- Шаблон каталога курса

Выполнение лабораторной работы

Установка git и gh командой dnf install

```
[yepolyakov@yepolyakov ~]$ sudo -i
[sudo] password for yepolyakov:
[root@yepolyakov ~]# 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@yepolyakov ~]# dnf install gh
Updating and loading repositories:
Repositories loaded.
Package
Installing:
gh
Arch
x86_64
Version
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-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-2.65.0-1.fc41.x86_64
Complete!
```

Рис. 1: Установка ПО

Базовая настройка git. Задаем имя и email, настраиваем utf-8, задаем имя начальной ветки и параметры autocrlf и safecrlf

```
[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: Базовая настройка

Создание ключей ssh по алгоритмам rsa и ed25519

```
[root@yepolykova1 ~]# 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's fingerprint is:
SHA256:CttdpwXrWwXidHwWdP-9uGyHtLSJt9PDL3J5K16tM root@yepolykova1.yepolykova1.net
The key's randomart image is:
+--[RSA 4096]-----+
|  + . 0 |
| +   + |
| +   + |
| . 0 . |
| . X . S o |
| o @ B.o . . |
| + * Xo*+o |
| | o +=*+E |
| omo.. |
+-----[SHA256]-----+

[root@yepolykova1 ~]# 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's fingerprint is:
SHA256:WuHTYCrucsgM1+KX/Gy8r4dIAWE1rsrByglJwM8 root@yepolykova1.yepolykova1.net
The key's randomart image is:
+--[ED25519 256]---+
| o+o  |
| ..  |
| ..  |
| ..  |
| o . . +. |
| omo o S.. |
| +o. E.o.. |
| ooo+ *+ |
| o.. +o+ |
| ..  |
+-----[SHA256]-----+

[root@yepolykova1 ~]#
```

Рис. 3: Создание ключей ssh

Создание ключа gpg

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

```
[root@polyakov:~]# gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code Group
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)
(5) ECC (sign and encrypt) "default"
(6) ECC (sign only)
(7) 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
  --w= key expires in n weeks
  --m= key expires in n months
  --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: yuliyapolakov
Email address: yuliyapolakov.87@mail.ru
Comment:
You selected this USER ID:
"yuliyapolakov <yuliyapolakov.87@mail.ru>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

Рис. 4: Создание ключа gpg

Добавление 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

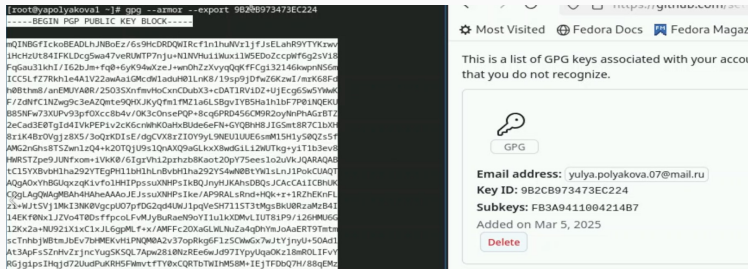


Рис. 5: Добавление ключа gpg

Настройка автоматических подписей коммитов git

```
[root@yapolyakoval ~]# git config --global user.signingkey 9B2CB973473EC224  
[root@yapolyakoval ~]# git config --global commit.gpgsign true  
[root@yapolyakoval ~]# git config --global gpg.program $(which gpg2)
```

Рис. 6: Настройка подписей коммитов

Клонирование репозитория в папку os-intro

Авторизуемся 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

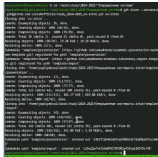
A terminal window showing the execution of a git clone command. The command is `git clone https://github.com:os-intro/os-intro.git`. The output shows the progress of cloning the repository, including the number of objects being transferred and the total size of the repository. The final output is `Cloning into 'os-intro'...` followed by a green prompt character.

Рис. 7: Клонирование репозитория в папку `os-intro`

Настройка каталога курса

Настройка каталога курса (удаление лишних файлов, создание необходимых каталогов командой make)

```
yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$ cat package.json
{
  "name": "os-intro",
  "version": "1.0.0",
  "description": "Операционные системы",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "os-intro"
  ],
  "author": "yepolyakova",
  "license": "MIT"
}

yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

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

yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$ make list
net-admin      Администрирование локальных сетей
net-os-admin   Администрирование сетевых подсистем
asch-pc        Архитектура IBM
sciprogram     Введение в научное программирование
netcybersec    Защита сетей и кибербезопасность
slawod         Математическое 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
sciprogram     Научное программирование
os-intro       Операционные системы
os2            Основы администрирования операционных систем
infosec-intro  Основы информационной безопасности

yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$ make prepare
yepolyakova@yepolyakova: ~/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 -dfx'
Entering 'template/presentation'
HEAD is now at c9b2712 Merge branch 'release/1.0.4'
Entering 'template/report'
HEAD is now at c26e22e Merge branch 'release/1.0.5'
yepolyakova@yepolyakova: ~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 8: Настройка каталога курса

Отправление файлов на сервер

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

```
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc_fignos/___init___py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc_fignos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc_fignos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc_fignos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
yepolyakova@yepolyakova:~/work/study/2024-2025/Операционные системы/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% (30/30), 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
yepolyakova@yepolyakova:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 9: Отправление файлов на сервер

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.

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