

Отчёт по лабораторной работе №2

Управление версиями

Гейдарова Назиля Рамиль кызы

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	17
4	Контрольные вопросы	18

Список иллюстраций

2.1	Загрузка пакетов	7
2.2	Параметры репозитория	8
2.3	rsa-4096	9
2.4	ed25519	10
2.5	GPG ключ	11
2.6	GPG ключ	12
2.7	Параметры репозитория	13
2.8	Связь репозитория с аккаунтом	14
2.9	Загрузка шаблона	15
2.10	Первый коммит	16

Список таблиц

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.

```
nrgeidarova@nrgeidarova:~$ git
использование: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
                [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
                [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
                [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
                [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
                <command> [<args>]
```

Стандартные команды Git используемые в различных ситуациях:

создание рабочей области (смотрите также: `git help tutorial`)

<code>clone</code>	Клонирование репозитория в новый каталог
<code>init</code>	Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: `git help everyday`)

<code>add</code>	Добавление содержимого файла в индекс
<code>mv</code>	Перемещение или переименование файла, каталога или символической ссылки
<code>restore</code>	Восстановление файлов в рабочем каталоге
<code>rm</code>	Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: `git help revisions`)

<code>bisect</code>	Выполнение двоичного поиска коммита, который вносит ошибку
<code>diff</code>	Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
<code>grep</code>	Вывод строк, соответствующих шаблону
<code>log</code>	Вывод истории коммитов
<code>show</code>	Вывод различных типов объектов
<code>status</code>	Вывод состояния рабочего каталога

Рис. 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.

```
nrgeidarova@nrgeidarova:~$  
nrgeidarova@nrgeidarova:~$ git config --global user.name "nrgeidarova"  
nrgeidarova@nrgeidarova:~$ git config --global user.email "1932234822@rudn.university"  
nrgeidarova@nrgeidarova:~$ git config --global core.quotepath false  
nrgeidarova@nrgeidarova:~$ git config --global init.defaultBranch master  
nrgeidarova@nrgeidarova:~$ git config --global core.autocrlf input  
nrgeidarova@nrgeidarova:~$ git config --global core.safecrlf warn  
nrgeidarova@nrgeidarova:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи


```

nrgeidarova@nrgeidarova:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nrgeidarova/.ssh/id_rsa):
Created directory '/home/nrgeidarova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nrgeidarova/.ssh/id_rsa
Your public key has been saved in /home/nrgeidarova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:r2Tuh5Eq5aLeCuBTzyo+xAkIY8maG20CxKtgMaexB88 nrgeidarova@nrgeidarova
The key's randomart image is:
+---[RSA 4096]-----+
|+..                |
|+X .               |
|*+%                |
|0=oE               |
|B=0.   S.         |
|=+. 0 . 0.        |
|00  = .00.        |
| +..+ 0+...       |
|.0*+.0 .+.        |
+----[SHA256]-----+
nrgeidarova@nrgeidarova:~$

```

Рис. 2.3: rsa-4096

```

nrgeidarova@nrgeidarova:~$
nrgeidarova@nrgeidarova:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/nrgeidarova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nrgeidarova/.ssh/id_ed25519
Your public key has been saved in /home/nrgeidarova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:uPTn0E00k7fZU3AoC73ikCOJNYStYrqkxprMna9FdZI nrgeidarova@nrgeidarova
The key's randomart image is:
+--[ED25519 256]--+
|  o                |
| . o               |
|  o   .   .       |
|... o  .o o o .   |
|o. o +oES= + o    |
|... +.=oX +   .   |
|+.  o.=.+.+ .    |
|+=. o  ++o o     |
|=o +o.  .o. .    |
+----[SHA256]-----+
nrgeidarova@nrgeidarova:~$ █

```

Рис. 2.4: ed25519

Создаем GPG ключ

```

nrgeidarova@nrgeidarova:~$ 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/nrgeidarova/.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)
Срок действия ключа не ограничен
Все верно? (y/N) y

```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

nrgeidarova@nrgeidarova:~$ 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   rsa4096/5701DF565EBEC3A4 2025-02-13 [SC]
      D31BFCE659FFB969755ADC0E5701DF565EBEC3A4
uid           [ абсолютно ] nrgeidarova <1132243103@rudn.university>
ssb   rsa4096/A0F6A11CA1D908A0 2025-02-13 [E]

nrgeidarova@nrgeidarova:~$ gpg --armor --export 5701DF565EBEC3A4
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGet9IoBEACydlyyr59pNNOXaxXEGTKY+c0MRRlGUHQ0ChUuj0rF+sN0Q5dV
Qrvpgmq0nXYVzYlMf8ty75eN8Sra7wKga9JhY89ezh0ZAUd0GFNL4sAQgRKvD
cvaTd81RIN50XnJ8UMC9q+Qljyq7gqVTJwrfMbTKUdWEf+UoAB5WOREI+L2c60k
9B0IHijGcEY9Dij0+pR6DtVtLnPXfs91vH0TwfORTHsm/m/vfrSku9665SXJfAVD
nWuAa0miQWRf6dR2e4nATeDVEi1AL70QP1PWJh5FjE+nIa2d+GkuSqm+pbu8N7ma
ECHYTYm2R/5apIDwjMnEKDPkmOnGlltdqp4RsoznXyTEm+FwoiY8/a3MzeLgQN8l
he1oKT7Mvb7ZXUeA0dAX/qaw/O6ikShNxPSlsRa0cfc62Ni3DsRzE1UFUTKD2jjY
NtDMb87P/tQGvGrZcFHZqL7bku8T/Cj748ZVKMvEXBFMYjhYapYj6zFSuQdGh5Ti
hXRmvfp8bphL3Nr70bQwjsjw3V3G3kKHgP/y6zAbU/QBB2hMpf13MsnQdbbTMxiw
b0Li40TY0fiEU71102fVzue+10iEYGN0SfL0D0ph0xug0Zv05/55ev0eN50F5+2o

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```

Fq3bQn2sInAHh38ENhu4Z9S7X3Y1+N2fk1H8L6KKl+Xm1dEVHCh+Edtpeo909FhE
umGK/mChbGsCykRKru4Ydh/k4AQTs7SudrmpD+m8rbgz8znjqk3qLKZwT4QuMwCV
/I+dEveNWYU400D6jNTdv1Klo4vQ4MRbBY0yyNEXavryoPhFPxFZFyrIre5qpGPA
Y08Wr7GpTmAENRB0gfflz05d90KcKQpFVo/0fob/Et4Qq90oeK4Sz9Wah88Dicoe
Tvw1jFAXdzTcX/IBKukJAbk/BiVpec09ehUw9sLKw2PQIEUpYnEj+jkcvM1Hph3b
0ZSyNpdeA==
==+8Xc
-----END PGP PUBLIC KEY BLOCK-----
nrgeidarova@nrgeidarova:~$
nrgeidarova@nrgeidarova:~$
nrgeidarova@nrgeidarova:~$
nrgeidarova@nrgeidarova:~$
nrgeidarova@nrgeidarova:~$ git config --global user.signingkey 5701DF565EBEC3A4
nrgeidarova@nrgeidarova:~$ git config --global commit.gpgsign true
nrgeidarova@nrgeidarova:~$ git config --global gpg.program $(which gpg2)
nrgeidarova@nrgeidarova:~$

```

Рис. 2.7: Параметры репозитория

Настройка gh

```
nrgeidarova@nrgeidarova:~$  
nrgeidarova@nrgeidarova:~$ 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/nrgeidarova/.ssh/id_rsa.pub  
? Title for your SSH key: GitHub CLI  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: 7272-8A45  
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/nrgeidarova/.ssh/id_rsa.pub  
✓ Logged in as nrgeidarova  
nrgeidarova@nrgeidarova:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
nrgeidarova@nrgeidarova:~$  
nrgeidarova@nrgeidarova:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"  
nrgeidarova@nrgeidarova:~$ cd ~/work/study/2024-2025/"Операционные системы"  
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=y  
amadharma/course-directory-student-template --public  
✓ Created repository nrgeidarova/os-intro on GitHub  
https://github.com/nrgeidarova/os-intro  
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com  
:nrgeidarova/os-intro.git os-intro  
Клонирование в «os-intro»...  
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJhbpZ1sF/zLDA0zPMSvHdKr4UvC0qU.  
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.
```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
nrgeidarova@nrgeidarova:~$  
nrgeidarova@nrgeidarova:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"  
nrgeidarova@nrgeidarova:~$ cd ~/work/study/2024-2025/"Операционные системы"  
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=y  
amadharma/course-directory-student-template --public  
✓ Created repository nrgeidarova/os-intro on GitHub  
https://github.com/nrgeidarova/os-intro  
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com  
:nrgeidarova/os-intro.git os-intro  
Клонирование в «os-intro»...  
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdKr4UvC0qU.  
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.
```

Рис. 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- хранилище - пространство на накопителе где расположен репозиторий
- commit - сохранение состояния хранилища
- история - список изменений хранилища (коммитов)
- рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

4. Опишите действия с VCS при единоличной работе с хранилищем.

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

10. Как и зачем можно игнорировать некоторые файлы при `commit`?

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить: