

Отчёт по лабораторной работе №2

Управление версиями

Анастасия Новикова

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.

```
aaanovikova@aaanovikova:~$ 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>]
```

Рис. 2.1: Загрузка пакетов

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

```
aanovikova@aanovikova:~$  
aanovikova@aanovikova:~$  
aanovikova@aanovikova:~$ git config --global user.name "aa-novikova"  
aanovikova@aanovikova:~$ git config --global user.email "1132246839@rudn.university"  
aanovikova@aanovikova:~$ git config --global core.quotepath false  
aanovikova@aanovikova:~$ git config --global init.defaultBranch master  
aanovikova@aanovikova:~$ git config --global core.autocrlf input  
aanovikova@aanovikova:~$ git config --global core.safecrlf warn  
aanovikova@aanovikova:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи


```

aanovikova@aanovikova:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aanovikova/.ssh/id_rsa):
Created directory '/home/aanovikova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aanovikova/.ssh/id_rsa
Your public key has been saved in /home/aanovikova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:tAPfG7G162LwGwW4s4770gBxA+haAu+Fqe6tyEh5fTI aanovikova@aanovikova
The key's randomart image is:
+---[RSA 4096]-----+
| ...                |
| 0 . 0              |
| 0. = .. . . .      |
| ..* . + 0 + .      |
| .= 0    So+ .       |
| 0 0 0    =.+0 .     |
| .0 . E . B.o.       |
| =.0    =.. =.       |
| ++..  +*0. ..       |
+---[SHA256]-----+
aanovikova@aanovikova:~$

```

Рис. 2.3: rsa-4096

```

aanovikova@aanovikova:~$
aanovikova@aanovikova:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aanovikova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aanovikova/.ssh/id_ed25519
Your public key has been saved in /home/aanovikova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:mxIeb29IZiu0LKKZ9jjFrSiYhpIzXu7wNC5Tap+ahuo aanovikova@aanovikova
The key's randomart image is:
+--[ED25519 256]--+
|
|
|
|
| . .o S
| . o.o++o
| *==o =o==o
| /@+*= +oo..
| %E0=.. . .
+----[SHA256]-----+
aanovikova@aanovikova:~$

```

Рис. 2.4: ed25519

Создаем GPG ключ

```
aanovikova@aanovikova:~$  
aanovikova@aanovikova:~$ 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/aanovikova/.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
```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

aanovikova@aanovikova:~$ 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   rsa4096/3E725CCC316F0372 2025-02-19 [SC]
      DF68E521884AADC4DE5F16143E725CCC316F0372
uid           [ абсолютно ] aa-novikova <1132246839@rudn.university>
ssb   rsa4096/F51930E7523C18BF 2025-02-19 [E]

aanovikova@aanovikova:~$ 
aanovikova@aanovikova:~$ gpg --armor --export 3E725CCC316F0372
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGe1zVoBEADKuM5XfqnlBd+gdbMTrs0GDUDW04eMC3bW+BI0s8mNEp5jaPT
Ax6RotVz+vytj5vzZkuIpJkztdiTLlAxckmkAWyID8Gy9tefr1pciYdkIp+QDEaF
E8wRJOc+9YFRjiWIPCVZlRxx3zApPYrsh2eIanZCcdcNnZBk0myraUdLUipT4nDs
JEHgA9Ue2Er+AGJhcAXyjU/FWEEqZOYCYJXkANUpWTHC2A4lHtMDCQ9qDsCRHFx8
AvxkBC0pVaFJ24DSF+7aSOZfcQrRSKJ7EPzDsLv8xZmgavSkUZyCuQtLzDT9LAo1
GGkJuRB8C0r28+990USBRYZdTEukidQTb0G6ZSsBPMfh6tdEgBpwTRtXcbsnGBej
eUoe3gUCNGRteFyo/uWmDjkhhy03Zd/sY28G0PeuolqLtxpdkg0Q5YZDp7+vLQQo
0w1TVvVjxLUXWyoGQS1d0Dcg0NDbPz/1GEW7mucID88/kC+FIBxG473EA4g3FXm
uY8cBAaCy/9/bJkdGjgA8KPP651AtDNwJcm2+9jC/LX3Xkps3+Mwz1vdTL0TLvM/

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
ZyIC1sKU3o6bHk9LW044b56BjW0sW8m5pmqgPMbe/fJk680rtE3HkhrIwspjrPcA  
c/0DsF5HlMAjxpfPe23QPLG3W0xWT6Bka7t11UjEb3d3LmVELJICbsRSmsWAZhKS  
TZH6HtWeA63pnqTqNZQCuIcZcTmCpq/bbkFdKCY2ZqkcGJa/a9ocZCcBa0XwRqj  
A+csmAyf7QTkepvnk8hu2jglfiVEya0qkLDVx6f38QmbSp00lykz3JkxDx847vHG  
CQ4rQpnyDndL9NvK14oiLV72qQxqwHarFk63bf3D9r4Qo4KTxYVllb5SebvrWC  
aQhRUprcGFuRUrJ4267XXs++9f0slo2n2sqfQwG9cHOMFJ1qfaufIKPitWqx+Na  
JwzXppqQ8Q==  
=m5C2
```

-----END PGP PUBLIC KEY BLOCK-----

```
aanovikova@aanovikova:~$  
aanovikova@aanovikova:~$ git config --global user.signingkey 3E725CCC316F0372  
aanovikova@aanovikova:~$ git config --global commit.gpgsign true  
aanovikova@aanovikova:~$ git config --global gpg.program $(which gpg2)  
aanovikova@aanovikova:~$
```

Рис. 2.7: Параметры репозитория

Настройка gh

```
aanovikova@aanovikova:~$ 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/aanovikova/.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: B2EB-A554
Press Enter to open https://github.com/login/device in your browser...
restorecon: SELinux: Could not get canonical path for /home/aanovikova/.mozilla/firefox/* restorecon: No such file or directory.
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/aanovikova/.ssh/id_rsa.pub
✓ Logged in as aa-novikova
aanovikova@aanovikova:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```

aanovikova@aanovikova:~$
aanovikova@aanovikova:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
aanovikova@aanovikova:~$ cd ~/work/study/2024-2025/"Операционные системы"
aanovikova@aanovikova:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=yamadharma/course-director
y-student-template --public
✓ Created repository aa-novikova/os-intro on GitHub
https://github.com/aa-novikova/os-intro
aanovikova@aanovikova:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:aa-novikova/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:+D1Y3wvV6TuJJhpZisF/zLDA0zPMSvHdkr4UvCOql.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```

create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/image/kulyabov.jpg
create mode 100644 project-personal/stage6/presentation/presentation.md
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.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/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
aanovikova@aanovikova:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (37/37), 342.28 КБ | 2.30 МБ/с, готово.
Total 37 (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:aa-novikova/os-intro.git
   0da8a45..0ba96a8 master -> master
aanovikova@aanovikova:~/work/study/2024-2025/Операционные системы/os-intro$

```

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: