0— ## Front matter title: “Отчет по лабораторной работе №3” subtitle: “Markdown” author: “Виктория Тиграновна Бекназарова”

0.1 Generic otions

lang: ru-RU toc-title: “Содержание”

0.2 Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

0.3 Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs: english ## Fonts mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9 ## Biblatex biblatex: true biblio-style: “gost-numeric” biblatexoptions: - parentracker=true - backend=biber - hyperref=auto - language=auto - autolang=other* - citestyle=gost-numeric ## Pandoc-crossref LaTeX customization figureTitle: “Рис.” tableTitle: “Таблица” listingTitle: “Листинг” lofTitle: “Список иллюстраций” lotTitle: “Список таблиц” lolTitle: “Листинги” ## Misc options indent: true header-includes: -

- keep figures where there are in the text
  - # keep figures where there are in the text

1 Цель работы

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

2 Выполнение лабораторной работы

1. Зададим имя и email владельца репозитория

```
vtbeknazarova@dk4n68 ~ $ git config --global user.name "BeknazarovaVika"

vtbeknazarova@dk4n68 ~ $
vtbeknazarova@dk4n68 ~ $ git config --global user.email "vbeknazarovaphoto@gmail.com"
```

1.png

2. Настроим utf-8 в выводе сообщений git

```
vtbeknazarova@dk4n68 ~ $ git config --global core.quotepath false
```

2.png

3. Зададим имя начальной ветки

```
vtbeknazarova@dk4n68 ~ $ git config --global init.defaultBranch master
```

3.png

4. Параметр autocrlf

```
vtbeknazarova@dk4n68 ~ $ git config --global core.autocrlf input
```

4.png

5. Параметр safecrlf

```
vtbeknazarova@dk4n68 ~ $ git config --global core.safecrlf warn
```

5.png

6. По алгоритму rsa с ключем размером 4096 бит

```
vtbeknazarova@dk4n68 ~ $ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vN9dJaWkjKa4pN4ZTZ+BD8HaH+Gpm+uE/e2zfXGyZlA vtbeknazarova@dk4n68
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|       .         |
|        o .    . .|
|       + + = oEo |
|      . S B o.o .|
|       * X +. .oo|
|      = B =  . o+|
|     + = = o.o+..|
|   .o +.=.o.==o. |
+----[SHA256]-----+
```

6.png

7. По алгоритму ed25519

```
vtbeknazarova@dk4n68 ~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_ed25519):
/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:/22N/m42zMfEW6XlmGRInyt761GMPyC84tK/jXjo6ZY vtbeknazarova@dk4n68
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|             .   |
|           . o . |
|          . . =oo|
|        S  o +.O+|
|         .  + *o=|
|        ..+. o+*+|
|       ..E+o+.+*B|
|        =*o+o*=*=|
+----[SHA256]-----+
```

7.png

8. Генерируем ключ

```
vtbeknazarova@dk4n68 ~ $ gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 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.

Выберите тип ключа:
   (1) RSA и RSA (по умолчанию)
   (2) DSA и Elgamal
   (3) DSA (только для подписи)
   (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
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
```

8.png

9. Выводим список ключей и копируем отпечаток приватного ключа

```
vtbeknazarova@dk4n68 ~ $ 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:   4  подписанных:   0  доверие: 0-, 0q, 0n, 0m, 0f, 4u
/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/.gnupg/pubring.kbx
------------------------------------------------------------------
sec   rsa4096/19C755F440B4EA28 2023-02-22 [SC]
      671F3DDF12F2482EEED53D0A19C755F440B4EA28
uid              [  абсолютно ] BeknazarovaVika <vbeknazarovaphoto@gmail.com>
ssb   rsa4096/9B4C7BBEC1CD9608 2023-02-22 [E]

sec   rsa4096/0D262938928B76F1 2023-02-22 [SC]
      ABDF3F96F6F9981FA7CB28D00D262938928B76F1
uid              [  абсолютно ] BeknazarovaVika <vbeknazarovaphoto@gmail.com>
ssb   rsa4096/445AD6A80953BCD6 2023-02-22 [E]

sec   rsa4096/1DF9A4FDEA2DD923 2023-02-22 [SC]
      22735D8A8ED46548546CBC1B1DF9A4FDEA2DD923
uid              [  абсолютно ] BeknazarovaVika <vbeknazarovaphoto@gmail.com>
ssb   rsa4096/7EC12652D7051F62 2023-02-22 [E]

sec   rsa4096/99B8B5AB80E27C70 2023-02-22 [SC]
```

9.png

10. Скопируйте наш сгенерированный PGP ключ в буфер обмена

```
vtbeknazarova@dk4n68 ~ $ gpg --armor --export 19C755F440B4EA28 | xclip -sel clip
```

10.png

11. Используя введёный email, указываем Git применяя его при подписи коммитов

```
vtbeknazarova@dk4n68 ~ $ git config --global user.signingkey 19C755F440B4EA28
vtbeknazarova@dk4n68 ~ $ git config --global commit.gpgsign true
vtbeknazarova@dk4n68 ~ $ git config --global gpg.program $(which gpg2)
```

11.png

12. Авторизовываемся

```
vtbeknazarova@dk4n68 ~ $ gh auth login
? What account do you want to log into? GitHub.com
? You're already logged into github.com. Do you want to re-authenticate? Yes
```

12.png

13. Создаем репозиторий

```
vtbeknazarova@dk4n68 ~ $ mkdir -p ~/work/study/2022-2023/"Операционные системы"
vtbeknazarova@dk4n68 ~ $ cd ~/work/study/2022-2023/"Операционные системы"
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы $ gh repo create study_2022-2023_os-int
ro --template=yamadharma/course-directory-student-template --public
✓ Created repository BeknazarovaVika/study_2022-2023_os-intro on GitHub
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы $ git clone --recursive https://github.
com/BeknazarovaVika/os-intro.git
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КиБ | 2.82 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Updating files: 100% (23/23), готово.
```

13.png

14. Переходим в каталог курса

```
vtbeknazarova@dk4n68 ~ $ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
```

14.png

15. Удаляем лишние файлы

```
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/os-intro $ rm package.json
rm: невозможно удалить 'package.json': Нет такого файла или каталога
```

15.png

16. Создаем необходиые каталоги

```
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/os-intro $ echo os-intro > COURSE
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/os-intro $ make
```

16.png

17. Отправляем файлы на сервер

```
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/os-intro $ git add .
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -am 'feat(main):
make course structure'
[master fd55d27] feat(main): make course structure
 361 files changed, 100327 insertions(+), 14 deletions(-)
 create mode 100644 labs/README.md
 create mode 100644 labs/README.ru.md
 create mode 100644 labs/lab01/presentation/Makefile
 create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
```

17.png

```
vtbeknazarova@dk4n68 ~/work/study/2022-2023/Операционные системы/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), 343.03 КиБ | 2.49 МиБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:BeknazarovaVika/os-intro.git
```

18.png

3 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 - запушить 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

4 Выводы

Я изучила идеологию и применение средств контроля версий. Освоила умения по работе с git.