

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: *Архитектура компьютера*

Студент: Кузнецова А. С.

Группа: НБИбд-01-23

МОСКВА

2023 г.

Содержание

1. Цель работы
2. Теоретическое введение
3. Выполнение лабораторной работы
 - 3.1. Базовая настройка GitHub
 - 3.2. Создание SSH ключа
 - 3.3. Создание рабочего пространства и репозитория курса
на основе шаблона
4. Вывод

1. Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2. Теоретическое введение

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, произведенные разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно

ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

3. Выполнение лабораторной работы

3.1. Настройка GitHub

Создаю учётную запись на сайте <https://github.com/> и заполните основные данные.

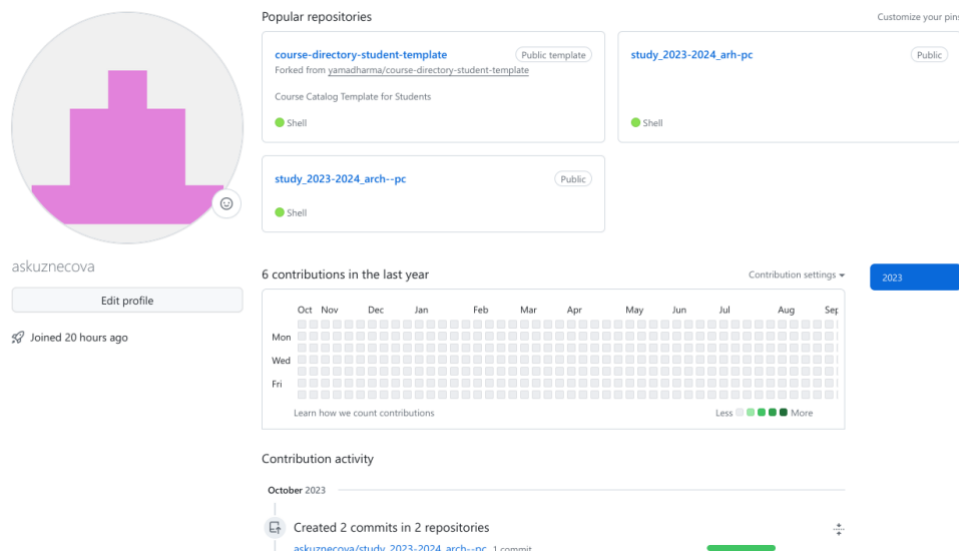


рис. 1

Создаем аккаунт в Github (рис. 1).

Делаем предварительную конфигурацию git. Открываем терминал и вводим следующие команды, указав свое имя и email (рис. 2).

```
askuznecova@dk8n78 ~ $ git config --global user.name "<Sasha Kuzetsova>"
askuznecova@dk8n78 ~ $ git config --global user.email "<aleksaschool2004@gmail.com>"
```

рис. 2

Настроим utf-8 в выводе сообщений git (рис. 3)

```
askuznecova@dk8n78 ~ $ git config --global core.quotePath false
```

рис. 3

Зададим имя начальной ветки (будем называть её master) (рис. 4)

```
askuznecova@dk8n78 ~ $ git config --global init.defaultBranch master
```

рис. 4

Параметр autocrlf (рис. 5)

```
askuznecova@dk8n78 ~ $ git config --global core.autocrlf input
```

рис. 5

Параметр safecrlf (рис. 6)

```
askuznecova@dk8n78 ~ $ git config --global core.safecrlf warn
```

рис. 6

3.2. Создание SSH ключа

Сгенерируем пару ключей (рис. 7)

```
askuznecova@dk8n78 ~ $ ssh-keygen -C "Sasha Kuznetsova <aleksaschool2004@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/a/s/askuznecova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/a/s/askuznecova/.ssh/id_rsa already exists.
Overwrite (y/n)? yes
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/a/s/askuznecova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/a/s/askuznecova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4qMxkgg6Dv70m/9odLVAeMjSJ1X8s3cms5C1FMgsCgU Sasha Kuznetsova <aleksaschool2004@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
|  Eo+ ++oo      |
|  .. *.++..    |
|  ...=.  ..    |
|  .   .  ..o.   |
|  .   . S o..+o |
|o. . . o . .+.o.o|
|= o.o + .   ..=.|
|+....+.o.    .  |
| o...++o..    |
+---[SHA256]-----+
```

рис. 7

Далее необходимо загрузить сгенерированный открытый ключ. Зайдём на сайт <http://github.org>. Скопировав из локальной консоли ключ в буфер обмена (рис. 8)

```
askuznecova@dk8n78 ~ $ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDxCccVvwzQj2DwyTpGESMiR7oizk2pkWPUzZWm1VbODEamkqfu4A2J7oebiQnxil+30aa7p1kT
4x5CeLlOjos/souVeaQH5GQ5VB3nocWQYNWmL8N3Xd1cGmtrTTdeh8Qx0UiDGaX4yo29n18WGu7jWPU0/N1f001V2f0bki1eB5R/tc1U4nrjmJun
FF8rS/S+wUroDZxjo6HaEsMhqNVZHstaKERU1H0Fzja77tY8MOtqY7X4W212f0M0+5oG0+Wip9iY8jxFtloWXQG4wCEMz71/JWMdf1SYbX3EjK
ABiQp99qqN3GvGBPwFxiJzFaFbdLUDLpAbLg6weI4E61em8U8NGX8G1bye64zo19rgDdcqy1eSX9jU+GeCuXQXhgovN80vhwpXo0V6SujI0knFW
3Z2bd0rtkS7rtSh8Vq6fakmKWJDuvzFzP0non14ud4eU7Gwf4G6r9MySnhX1Cjj1/Q78B6MH4m1CBaqG8vPZniJDS4+3/yuQTApwfEU= Sasha K
uznetsova <aleksaschool2004@gmail.com>
```

рис. 8

9) Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (рис.

Authentication Keys

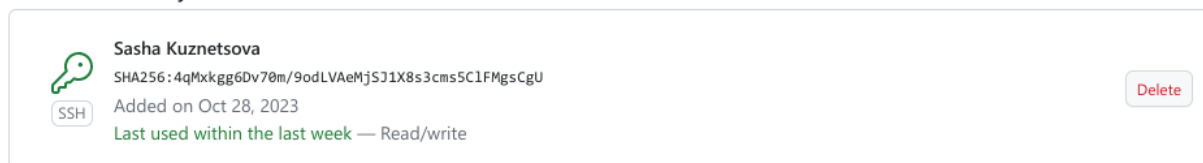


рис. 9

3.3. Создание рабочего пространства и репозитория курса на основе шаблона

Создадим каталог для предмета «Архитектура компьютера» (рис. 10)

```
askuznecova@dk8n78 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис. 10

Задав имя новому репозиторию, переходим в каталог курса (рис. 11)

```
askuznecova@dk8n78 ~ $ cd work/study/2023-2024/"Архитектура компьютера"
```

Рис. 11

Клонируем созданный репозиторий (рис. 12)

```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:askuznecova/study_2023-2024_arch--pc.git
Клонирование в «study_2023-2024_arch--pc»...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 28 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (28/28), 17.30 КиБ | 17.30 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/a/s/askuznecova/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 498.00 КиБ/с, готово.
```

Рис. 12

Перейдём в каталог курса (рис. 13)


```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/"Архитектура компьютера"/study_2023-2024_arch--pc
```

Рис. 13

Удалим лишние файлы (рис. 14)

```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc $ rm package.json
```

Рис. 14

Создадим необходимые каталоги (рис. 15)

```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc $ mkdir package.json
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc $ echo arch-pc > COURSE
E
```

Рис. 15

Запускаем команду «make» (рис. 16)

```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc $ make
```

Рис. 16

Отправляем файлы на сервер (рис. 17)

```
askuznecova@dk8n78 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arch--pc $ git commit -am "feat
(main): make course structure"
[main: de36ab2] feat(main): make course structure
199 files changed, 54725 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
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
```

Рис. 17

Проверяем правильность создания иерархии рабочего пространства в локальном репозитории и на странице github (рис. 18)

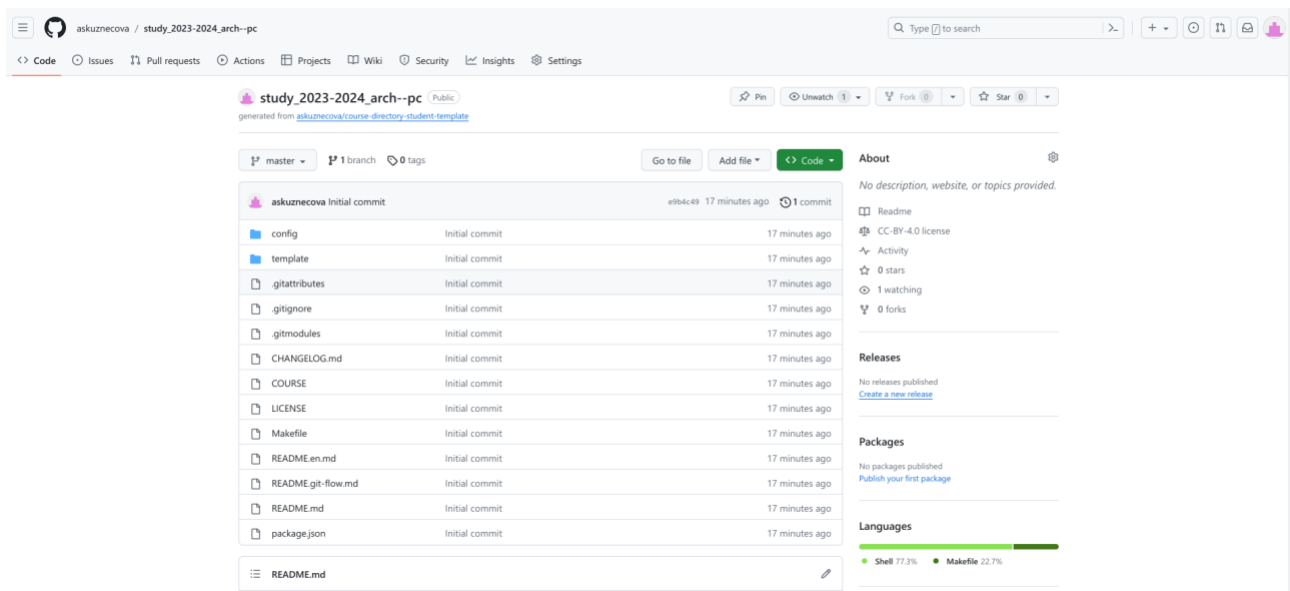


Рис. 18

4. Вывод

Я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой git.