

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: Карпова Е.А.

Группа: НКАбд-02-23

МОСКВА

2023 г.

Содержание

1 Цель работы.....	3
2 Задание	4
3 Теоретическое введение	5
4 Выполнение лабораторной работы.....	7

1 Цель работы

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

2 Задание

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение заданий для самостоятельной работы.

3 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, произведённые 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 изменения, сделанные разными участниками, 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. 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. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

4 Выполнение лабораторной работы

1 Настройка GitHub

Создаю учетную запись на сайте GitHub. Далее я заполнила основные данные учетной записи. Аккаунт создан (рис. 1)



Рис. 1. Создание учетной записи.

2 Базовая настройка Git

Открываю консоль и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней свою электронную почту (рис. 2)

```
eakarpova@dk8n61 ~ $ git config --global user.name "Esenia Karpova"
^[[Aeakarpova@dk8n61 ~ $
eakarpova@dk8n61 ~ $ git config --global user.email "1132236008@pfur.ru"
```

Рис. 2. Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 3).

```
eakarpova@dk8n61 ~ $ git config --global core.quotepath false
```

Рис. 3. Настройка utf-8

Задаю имя «master» для начальной ветки (рис. 4).

```
eakarpova@dk8n61 ~ $ git config --global init.defaultBranch master
```

Рис. 4. Задаю имя «master»

Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 5). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

```
eakarpova@dk8n61 ~ $ git config --global core.autocrlf input
```

Рис. 5. Задаю параметр «autocrlf»

Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость (рис. 6). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
eakarpova@dk8n61 ~ $ git config --global core.safecrlf warn
```

Рис. 6. Задаю параметр `safecrlf`

3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца (рис. 7). Ключ автоматически сохранится в каталоге `~/.ssh/`


```
eakarpova@dk8n61 ~ $ ssh-keygen -C "Esenia Karpova 1132236008@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/e/a/eakarpova/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/e/a/eakarpova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/a/eakarpova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/a/eakarpova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:T1jWk3inCH8sYQJBPr1570tKYhVqqIL94EBjAgX8yk Esenia Karpova 1132236008@pfur.ru
The key's randomart image is:
+----[RSA 3072]-----+
|. +o+|.
|o+ + o o|.
|o E * o * =|.
| B ..0 = +|.
| o ooo S +|.
| o= + o|.
| .o.o o|.
| .o.o...oo|.
| oo .....|
+----[SHA256]-----+
```

Рис. 7. Генерация SSH-ключа

Далее необходимо загрузить сгенерированный открытый ключ. Для этого зайти на сайт под своей учётной записью и перейти в меню Setting . После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key . Скопировав из локальной консоли ключ в буфер обмена (рис. 8). Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

```
eakarpova@dk8n61 ~ $ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC9AZaIvE4cg01HiwTzrvKfBRrCQ0KdhYQK2twvT5YfpnBRsSTcgTYBQuz/eknfChZ1GB6/QF
pn5kQjTQVBGBYF1Z/MBNLKNBbHI ZLGYoDruEQLDNSDh0vTm/NDC49HIxgkmtsZr27R1Hg1HLfgDYT37/g+aH7+WDdBiMpvGB9xfRsfPONpE7L
nquvt1IqINLukKQrEBoucShkZ88XyI14A7kUq5gHuDg6uBSsi7dTKzULMv4jWssRrzL5NU1aD3ZRcrz3j07uxS3PvdCmiE8P8h2DH0yUvAqRrQ
2dGZiYhwUfTek/3EacLtedQIbChetqcGRr3B/+WqwUNNVq/wwsik2nGKJ/R3YHmK35Kb4JKSssLLeM9ounUIMAXWTus+2dYUvGQsuWdub93rr6
dE8QHgxqy0c0DmvIfjpJSJeHTSj5KGI4uWOZQLzxF7aXdUSRuFb1aFjvwskZzMP8xE39U167L2YW03Ic6UxxKAE7s= Esenia Karpova 11
008@pfur.ru
```

Рис. 8. Копирование содержимого файла

4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -r создаю все директории после домашней ~/work/study/2022-2023/“Архитектура компьютера” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги.

5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 9).

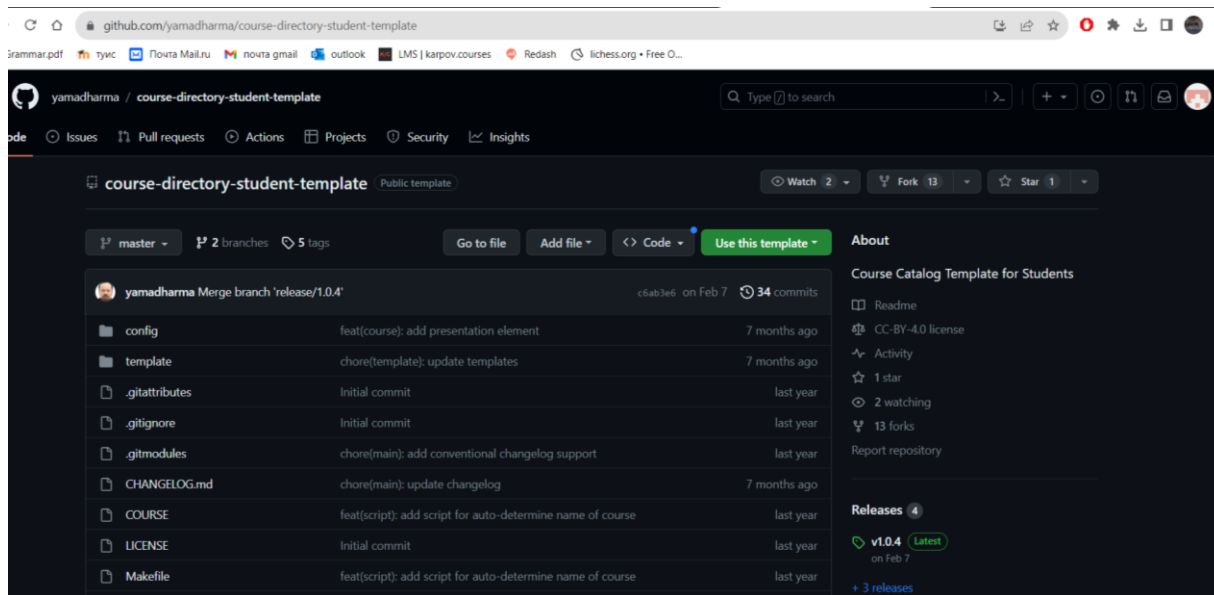


Рис. 9. Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study_2022–2023_arh-рс и создаю репозиторий, нажимаю на кнопку «Create repository from template». Репозиторий создан (рис. 10).

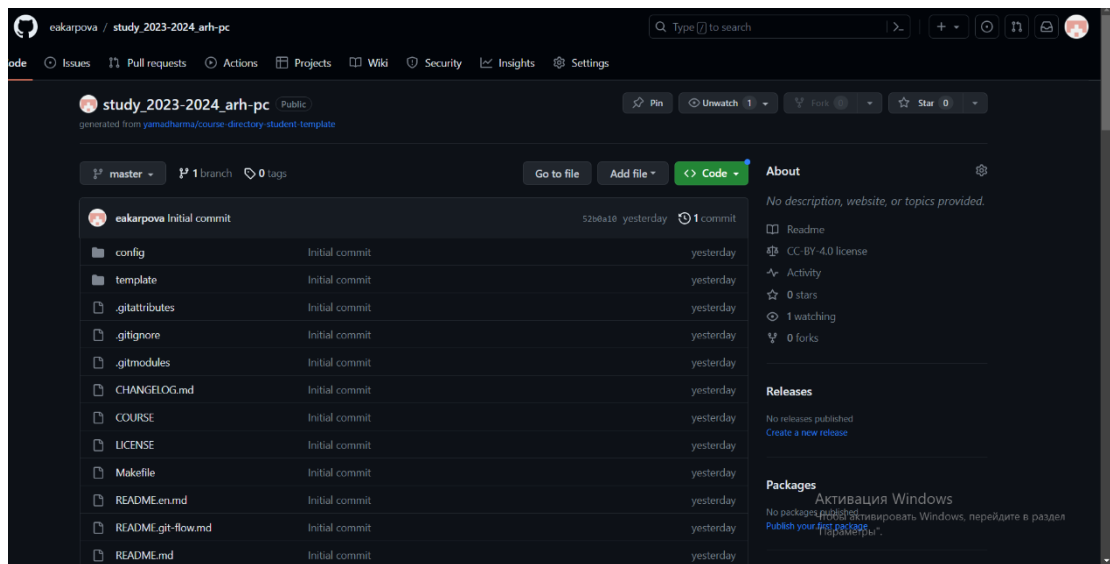


Рис. 10. Репозиторий создан

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd` (рис. 11).

```
eakarpova@dk5n58 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис. 11. Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com:study_2022-2023_arh-pc.git arch-pc` (рис. 12).

```
eakarpova@dk5n58 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:eakarpova/study_2023-2024_arh-pc.git
Клонирование в «study_2023-2024_arh-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/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.
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), 16.93 КиБ | 1.21 МБ/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/e/a/eakarpova/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-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 | 897.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/e/a/eakarpova/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 1.59 МБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 12. Клонирование репозитория

6 Настройка каталога курса

Перехожу в каталог курса

```
eakarpova@dk8n61 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/study_2023-2024_arh-pc
```

Рис. 13. Переход в каталог курса

Удаляю лишние файлы с помощью rm

```
eakarpova@dk8n61 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc $ rm pack
```

Рис. 14 Удаление лишних файлов

Создаю необходимые каталоги

```
eakarpova@dk8n61 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc $ echo arch-pc > COURSE
```

Рис. 15 Создание каталога

Отправляю файлы на сервер с помощью команд

```
eakarpova@dk8n61 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc $ git commit -am 'feat(main): make course structure'
[master 7894051] 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
```

Рис. 16 Отправка файлов

Проверяю правильность создания рабочего пространства на странице github

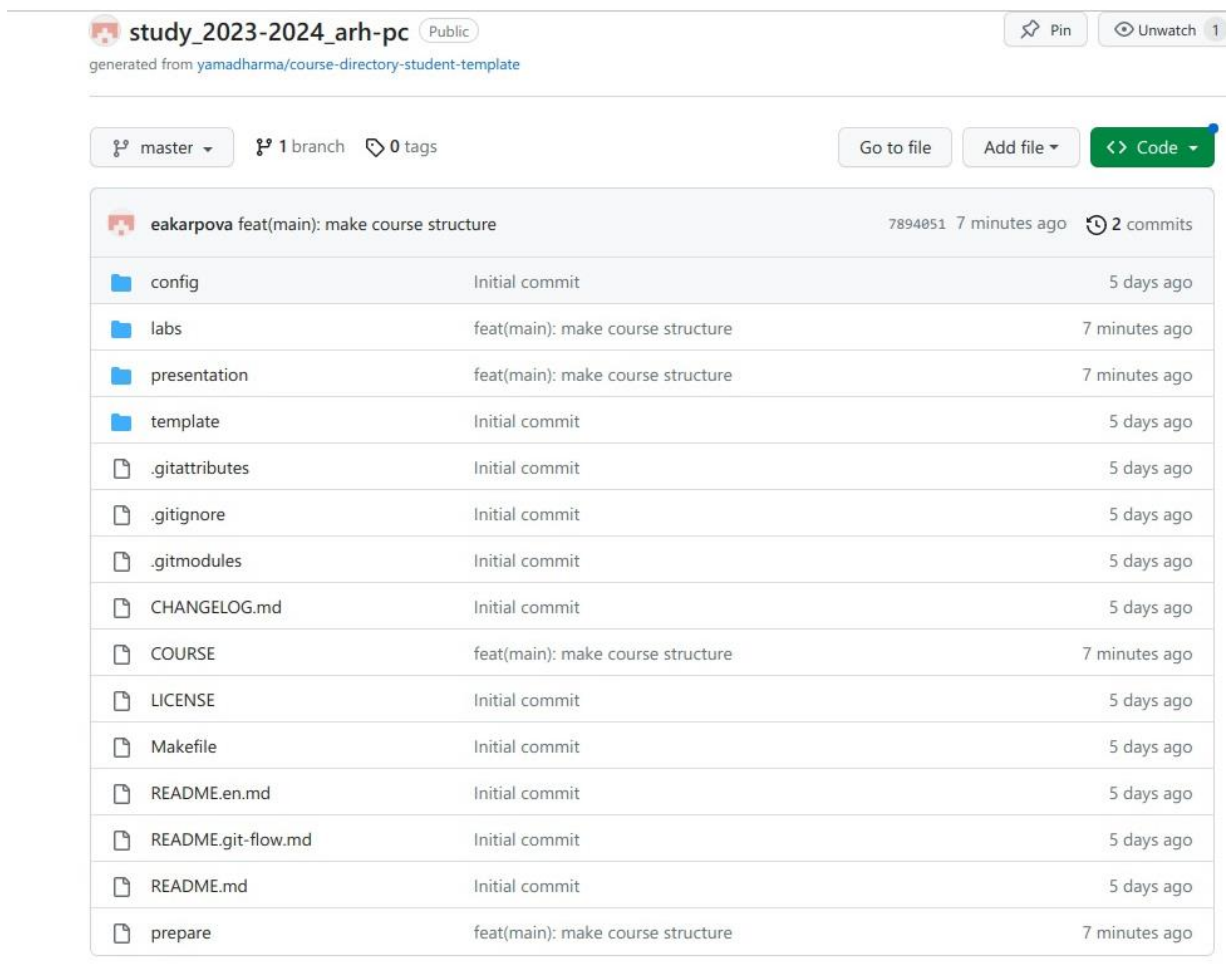


Рис. 17 Страница github

7 Задания для самостоятельной работы

С помощью изученных в ходе первой лабораторной работы команд помещаю отчет по первой лабораторной работе в папку `labs > lab02 > report`

Скопирую отчет по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства

Перемещаю отчет по первой лабораторной работе в `~/lab01/report`.

Загружаю файл на github

Ссылка:

https://github.com/eakarpova/study_2023-2024_arh-pc/tree/master

8 Вывод

В ходе лабораторной работы я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой

git

