Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Воронов Александр Валерьевич

Содержание

# 1 Цель работы

Целью работы является применение средств контроля версий. А также очень важно приобрести практические навыки по работе с системой git.

# 2 Задание

* Настройка GitHub.
* Базовая настройка Git.
* Создание SSH-ключа.
* Создание рабочего пространства и репозитория курса на основе шаблона.
* Создание репозитория курса на основе шаблона.
* Настройка каталога курса.
* Выполнение заданий для самостоятельной работы.

#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.

# 3 Выполнение лабораторной работы

## 3.1 Настройка Github

Для выполнения лабораторной работы создаю учетную запись на https://github.com/

##Базовая настройка git

Делаю предварительную конфигурацию git. Захожу в терминал и ввожу команды, указывая свое имя и email (рис. -fig. 1)

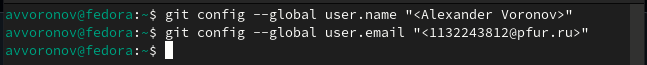


Рис. 1: Предварительная конфигурация git

Настраиваю utf-8 в выходе сообщений git.(рис. -fig. 2)

Настраиваю utf-8

Рис. 2: Настраиваю utf-8

Задаю имя начальной ветки, которую буду называть master (рис. -fig. 3)

Имя начальной ветки

Рис. 3: Имя начальной ветки

Ввожу autocrlf и safecrlf (рис. -fig. 4)

Использование autocrlf

Рис. 4: Использование autocrlf

(рис. -fig. 5)

Использование safecrlf

Рис. 5: Использование safecrlf

##Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев генерирую пару ключей (приватный и открытый). (рис. -fig. 6)

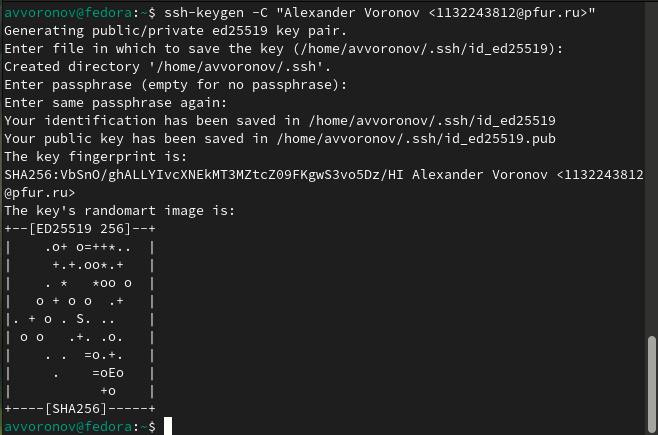


Рис. 6: Генерирование ключей SSH

Чтобы скопировать из локальной консоли ключ в буфер обмена, устанавливаю команду xclip. Затем воспользуюсь командой xclip (рис. -fig. 7)

Копирование SSH-ключа

Рис. 7: Копирование SSH-ключа

Вставляю ключ в появившееся на сайте поле, указываю его имя.(рис. -fig. 8)

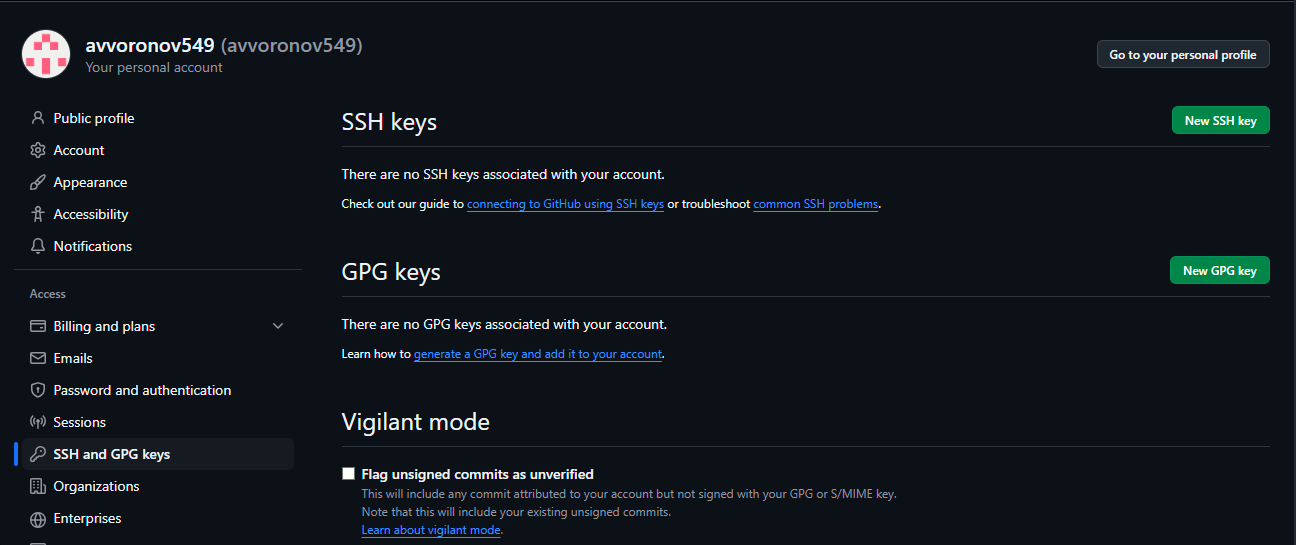


Рис. 8: Вставка SSH-ключа

##Создание рабочего пространства на основе шаблона

Открываю терминал и создаю репозиторий для предмета «Архитектура компьютеров». (рис. -fig. 9)

Создание репозитория

Рис. 9: Создание репозитория

##Создание репозитория курса на основе шаблона

Захожу на страницу репозитория с шаблоном курса, выбираю его в качестве своего нового. (рис. -fig. 10)

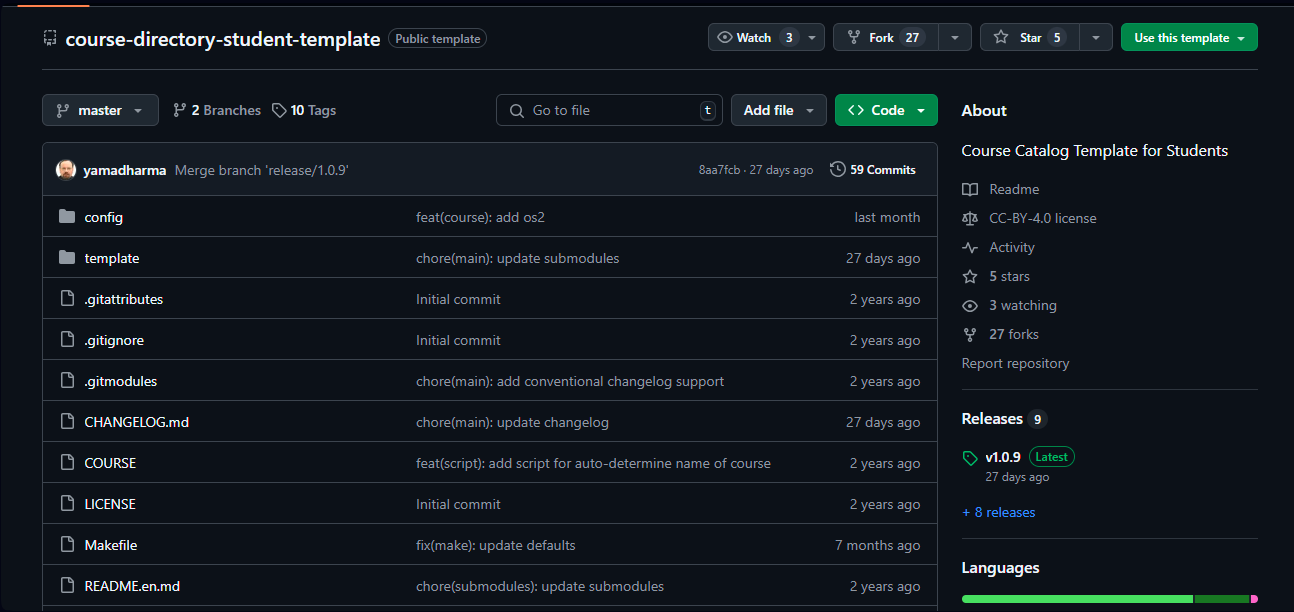


Рис. 10: Выбор шаблона

Затем создаю его, дав ему название. (рис. -fig. 11)

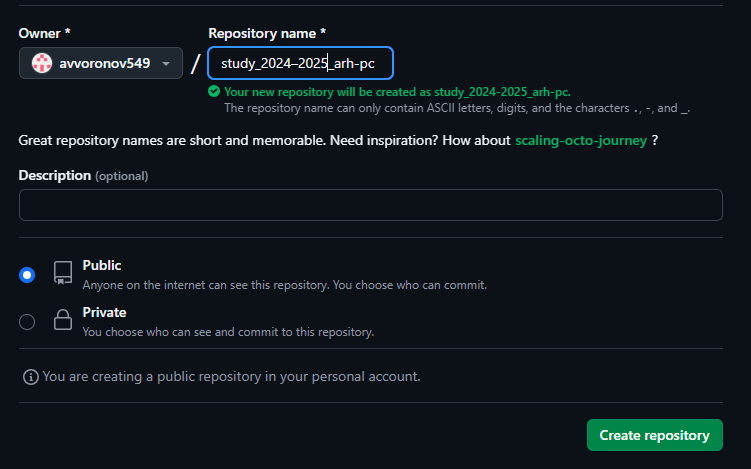


Рис. 11: Cоздание репозитория

Открываю терминал и перехожу в каталог курса. (рис. -fig. 12)

Переход в каталог курса

Рис. 12: Переход в каталог курса

Клонирую созданный репозиторий (рис. -fig. 13)

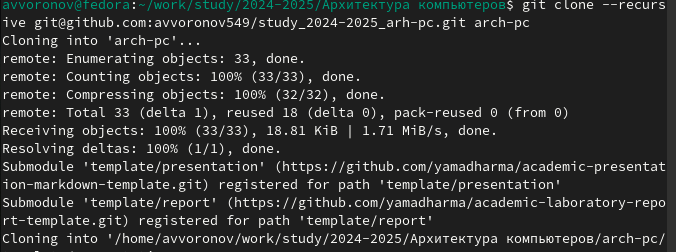


Рис. 13: Клонирование репозитория

##Настройка каталога курса

Перехожу в каталог курса и удаляю лишний файлы (рис. -fig. **¿fig:014?**)

[Переход в каталог курса] (image/017.PNG){#fig:014 width=70%}

(рис. -fig. 14)

Удаление лишних файлов в каталоге курса

Рис. 14: Удаление лишних файлов в каталоге курса

Создаю необходимые каталоги, отправляю файлы на сервер (рис. -fig. 15)

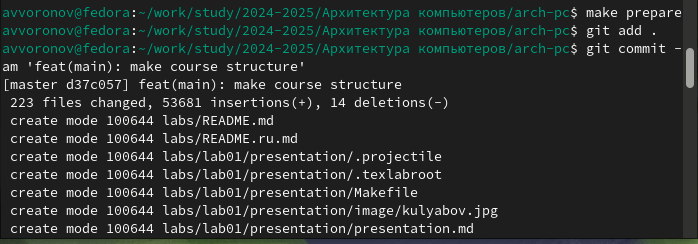


Рис. 15: Создание нужных каталогов и отправка на сервер

В репозитории проверяю результат выполненной работы. (рис. -fig. 16)

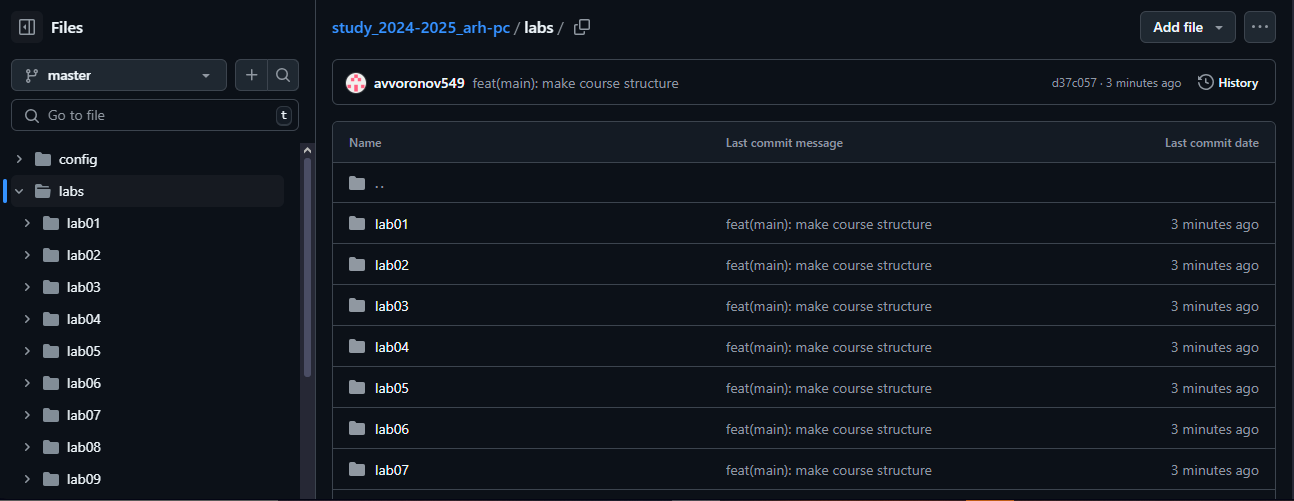


Рис. 16: Проверка проделанной работы

##Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch, после чего перехожу в директорию labs/lab01/report (рис. -fig. 17)

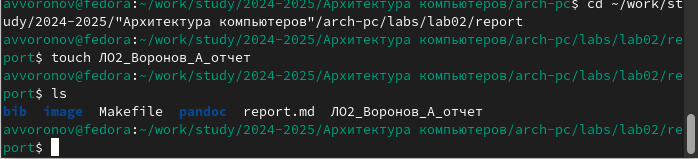


Рис. 17: Создание отчета

(рис. -fig. 18)

Переход в директорию labs/lab01/report

Рис. 18: Переход в директорию labs/lab01/report

Проверяю местонахождение файла с отчетом по первой лабораторной работой. Он должен быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls. (рис. -fig. 19)

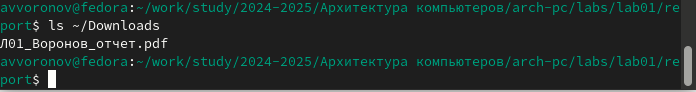


Рис. 19: Проверка расположения файла

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls. (рис. -fig. 20)

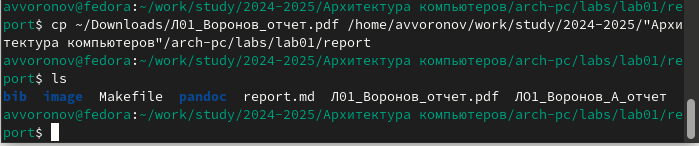


Рис. 20: Копирование файла

После чего добавляю файл в add Сохраняю изменения на сервере командой git commit -m “…”, поясняя, что добавил файлы (рис. -fig. 21)

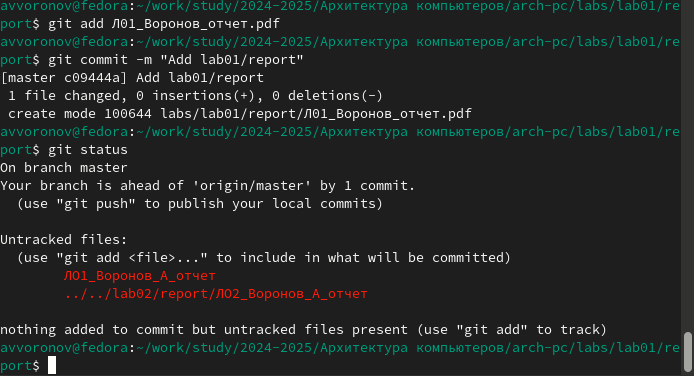


Рис. 21: Комментарии к изменениям

Аналогично я поступаю с отчётом по этой лабораторной работе, после чего отправляю коммит на сервер.(рис. -fig. 22)

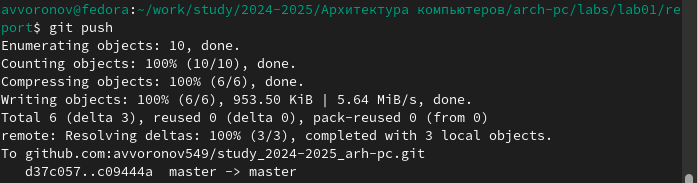


Рис. 22: Отправление на сервер

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. -fig. 23)

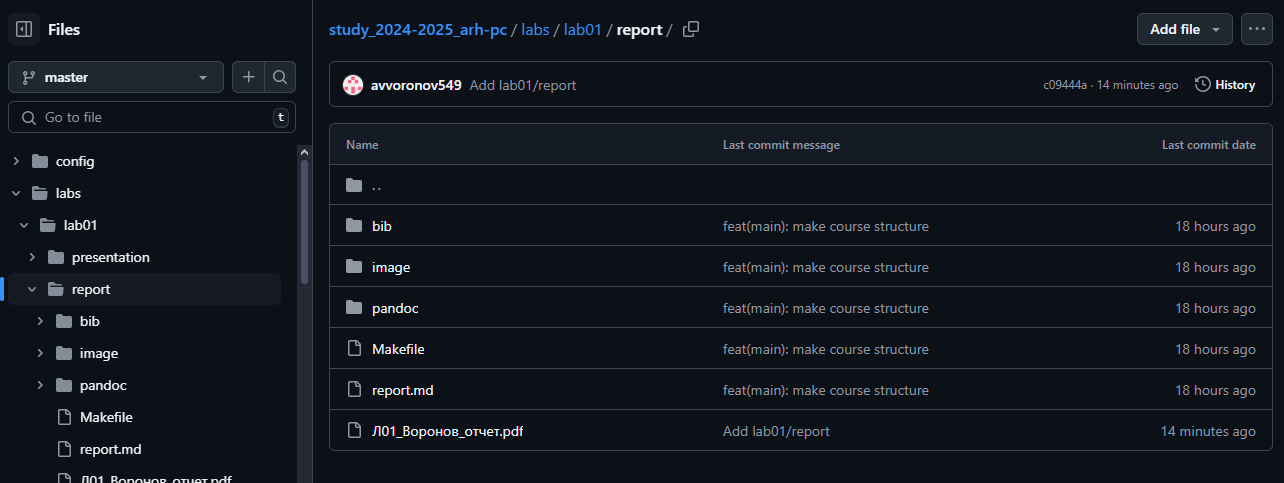


Рис. 23: Проверка на нахождения файла

# 4 Выводы

В заключение хочется отметить, что данная лабораторная работа позволила мне научиться работать с системой Git. Я практиковал свои навыки в работе с командной строкой, теперь уже связывая выполнимое с директориями GitHub.

# 5 Список литературы

1. Архитектура ЭВМ https://esystem.rudn.ru/pluginfile.php/2089082/mod\_resource/content/0/Лаборато рная%20работа%20№2.%20Система%20контроля%20версий%20Git.pdf
2. 30 команд Git, необходимых для освоения интерфейса командной строки Git / Хабр https://habr.com/ru/companies/ruvds/articles/599929/
3. Система контроля версий: определение, функции, популярные решения https://gb.ru/blog/sistema-kontrolya-versij/
4. Мой Github: https://github.com/avvoronov549/study\_2024-2025\_arh-pc