**Российский университет дружбы народов**

**Факультет физико-математических и естественных наук**

**Кафедра Компьютерные и информационные науки**

**Отчёт**

**По Лабораторной работе №2**

Дисциплина: Архитектура компьютера

Студент: Сахно Алёна Юрьевна

Студ.билет: 1132243813

Группа:НКАбд-04-24

Содержание

[Цель работы: 3](#_Toc178329639)

[2.2. Теоретическое введение 3](#_Toc178329640)

[2.2.1. Системы контроля версий. Общие понятия 3](#_Toc178329641)

[2.2.2. Система контроля версий Git 4](#_Toc178329642)

[3. Практическая часть 4](#_Toc178329643)

[3.1. Порядок выполнения лабораторной работы 4](#_Toc178329644)

[3.1.1. Настройка github 4](#_Toc178329645)

[3.1.2. Базовая настройка git 4](#_Toc178329646)

[3.1.3. Создание SSH ключа 5](#_Toc178329647)

[3.1.4. Сознание рабочего пространства и репозитория курса на основе шаблона 6](#_Toc178329648)

[3.1.5. Сознание репозитория курса на основе шаблона 7](#_Toc178329649)

[3.1.6. Настройка каталога курса 9](#_Toc178329650)

[4. Задание для самостоятельной работы 10](#_Toc178329651)

[Вывод: 12](#_Toc178329652)

# Цель работы:

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

# 2.2. Теоретическое введение

# 2.2.1. Системы контроля версий. Общие понятия

**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 (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.

Демидова А. В.

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.

# 2.2.2. Система контроля версий Git

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.

# 3. Практическая часть

# 3.1. Порядок выполнения лабораторной работы

# 3.1.1. Настройка github

Существует несколько доступных серверов репозиториев с возможностью бесплатного размещения данных. Например, http://bitbucket.org/, https://github.com/ и https://gitflic.ru. Для выполнения лабораторных работ предлагается использовать Github. Создайте учётную запись на сайте https://github.com/ и заполните основные данные.

# 3.1.2. Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:

git config --global user.name "< Name Surname>"

git config --global user.email "< work@mail>"

d:\Downloads\2024-09-23_20-59-57.png

Настроим utf-8 в выводе сообщений git:

git config --global core.quotepath false

d:\Downloads\2024-09-23_21-03-26.png

Зададим имя начальной ветки (будем называть её master):

git config --global init.defaultBranch master

d:\Downloads\2024-09-23_21-05-44.png

Параметр autocrlf:

git config --global core.autocrlf input

d:\Downloads\2024-09-23_21-07-42.png

Параметр safecrlf:

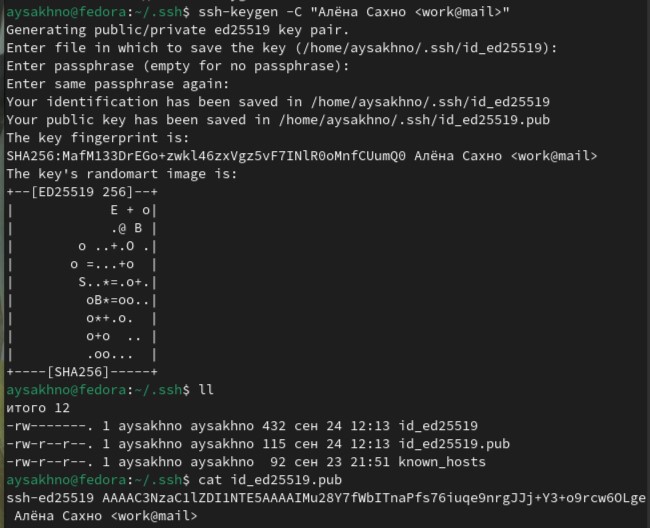
git config --global core.safecrlf warn

d:\Downloads\2024-09-23_21-09-59.png

# 3.1.3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый):

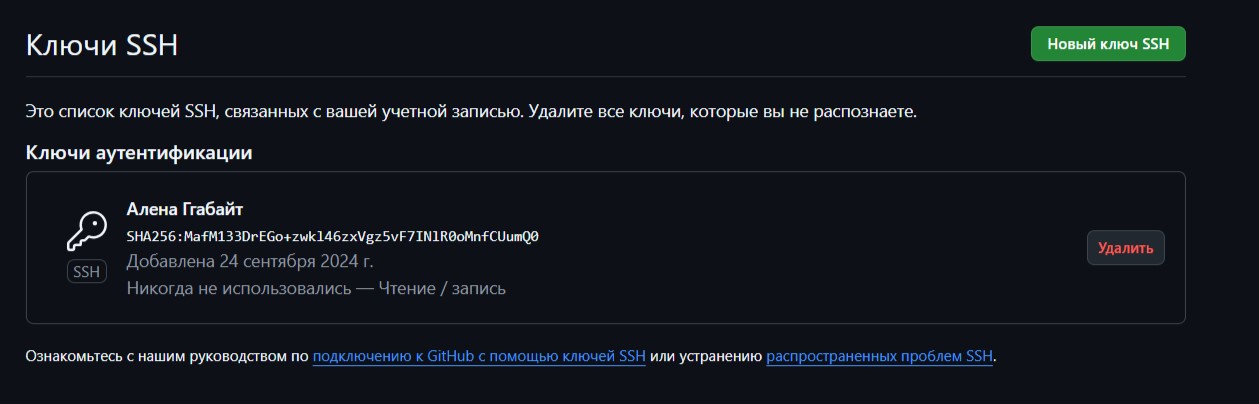
ssh-keygen -C "Имя Фамилия <work@mail>"



Ключи сохраняться в каталоге ~/.ssh/.

Далее необходимо загрузить сгенерённый открытый ключ. Для этого зайти на сайт http: //github.org/ 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

cat ~/.ssh/id\_rsa.pub | xclip -sel clip



вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

# 3.1.4. Сознание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Рабочее пространство по предмету располагается в следующей иерархии:

~/work/study/

└── < учебный год>/

└──<название предмета> /

└── < код предмета>/

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) структура каталогов примет следующий вид:

~/work/study/

└── 2023–2024/

└── Архитектура компьютера/

└── arch-pc/

└── labs/

└── lab01/

└── lab02/

└── lab03/ ...

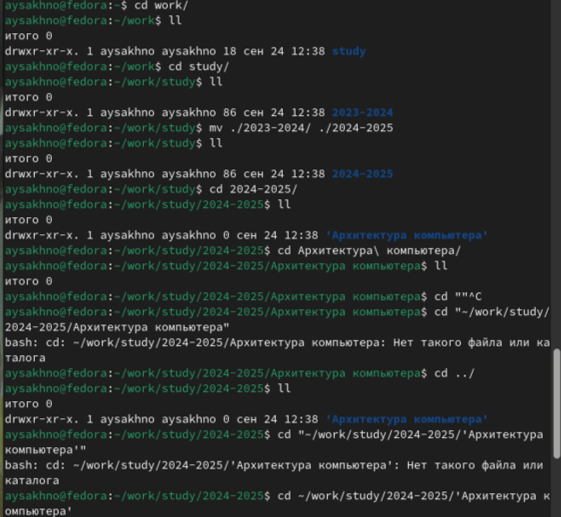
• Каталог для лабораторных работ имеет вид labs.

• Каталоги для лабораторных работ имеют вид lab, например: lab01, lab02 и т.д.

Название проекта на хостинге git имеет вид: study\_\_

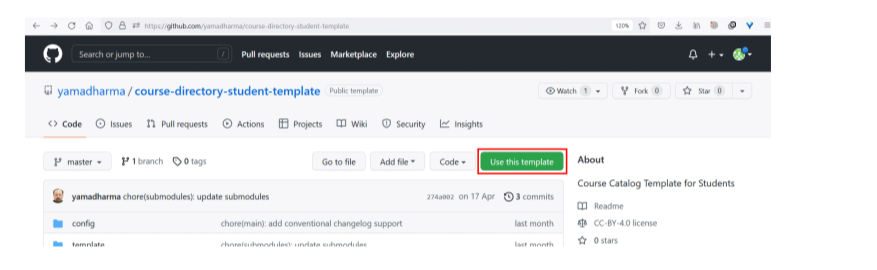
Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид: study\_2023–2024\_arch-pc Откройте терминал и создайте каталог для предмета «Архитектура компьютера»:

mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"



# 3.1.5. Сознание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github. Перейдите на станицу репозитория с шаблоном курса https://github.com/yamadharma/cour se-directory-student-template. Далее выберите Use this template.



В открывшемся окне задайте имя репозитория (Repository name) study\_2023–2024\_arhpc и создайте репозиторий (кнопка Create repository from template).

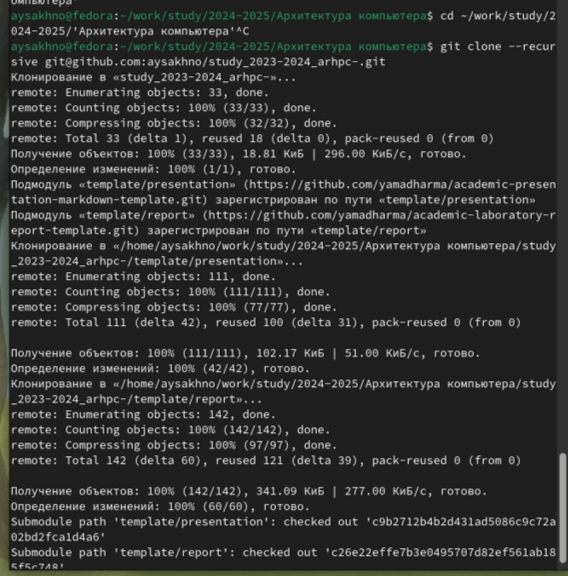
Откройте терминал и перейдите в каталог курса:

cd ~/work/study/2023–2024/"Архитектура компьютера"

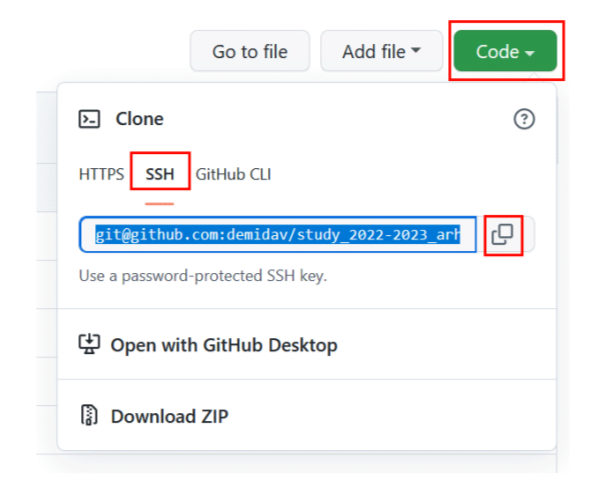
клонируйте созданный репозиторий:

git clone --recursive [git@github.com:/study\_2023–2024\_arh-pc.git](mailto:git@github.com:/study_2023–2024_arh-pc.git)

↪ arch-pc



Ссылку для клонирования можно скопировать на странице созданного репозитория Code -> SSH:



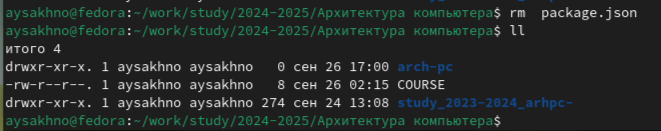
# 3.1.6. Настройка каталога курса

Перейдите в каталог курса:

cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc

Удалите лишние файлы:

rm package.json



Создайте необходимые каталоги:

echo arch-pc > COURSE

make

Отправьте файлы на сервер:

git add . git commit -am 'feat(main): make course structure'

git push

Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.

Вывод: Я изучила идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 4. Задание для самостоятельной работы

1. **Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).**

Перехожу в директорию labs/lab01/report с помощью утилиты cd. Создаю

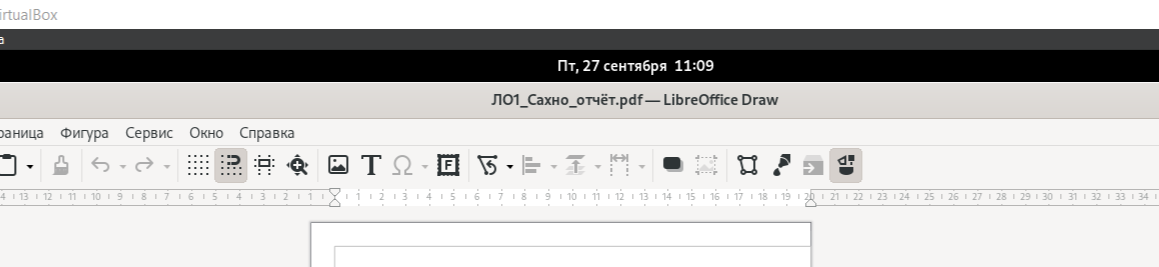
в каталоге файл для отчета по третьей лабораторной работе с помощью

утилиты touch

d:\Downloads\2024-09-26_23-51-39.png

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в

меню приложений. В нем я открыла файл и теперь могу в нем работать.



1. **Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие**

**каталоги созданного рабочего пространства.**

d:\Downloads\2024-09-27_00-01-52.png

Проверяю местонахождение файлов с отчетами по первой лабораторным работам. Они должны быть в подкаталоге домашней директории «Загрузки»,

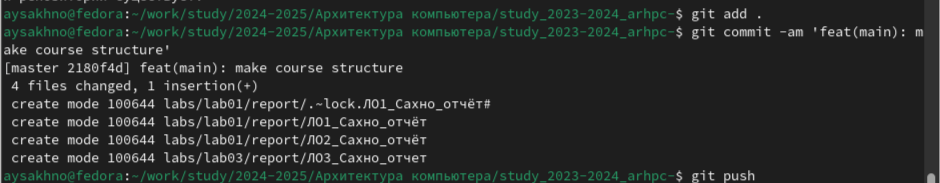
для проверки использую команду ls.

C:\Users\___\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\2024-09-27_01-43-26.jpeg

1. **Загрузите файлы на github.**

Добавляю с помощью команды git add в коммит созданные файлы:

Л01\_Сахно\_ отчёт



Сохраняю изменения на сервере командой git commit -m “…”, поясняя, что добавила файлы.

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается. При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам. Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report

C:\Users\___\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\2024-09-27_11-05-24.jpeg

Вывод:

Я изучила идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.