Лабораторная работа №2

Первоначальная настройка git

Емельянов Антон

Содержание

# 1. Цель работы

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

# 2. Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 3. Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в [табл. 1](#tbl-std-dir) приведено краткое описание стандартных каталогов Unix.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 1: Описание некоторых каталогов файловой системы GNU Linux   | Имя каталога | Описание каталога | | --- | --- | | / | Корневая директория, содержащая всю файловую | | /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям | | /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ | | /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя | | /media | Точки монтирования для сменных носителей | | /root | Домашняя директория пользователя root | | /tmp | Временные файлы | | /usr | Вторичная иерархия для данных пользователя | |

Более подробно про Unix см. в [1–4].

# 4. Выполнение лабораторной работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown. Переместимся в каталог 2 лабы Откроем файл report.qmd и начнём исправлять его, превращая его в отчёт по лабе 2 используем make, чтобы создать pdf и docx Затем загружаем файлы на github В ходе работы с языком разметки markdown были получены базовые навыки редактуры и обращения с отчётами Лабораторная работа №2 Первоначальная настройка git Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git. Начнём установку ПО с git: (**?@fig-001**)

Далее установим gh, а также внесём всю базовую информацию: имя глобального пользователя, адрес электронной почты, настроим utf-8 в выводе сообщений git. Зададим имя начальной ветки (master) и установим параметры autocrlf и safecrlf

(**?@fig-002**)

(**?@fig-003**)

Сгенерируем ключи размера 4096 по алгоритму RSA и по алгоритму ed25519, также сгенерируем pgp ключ, он понадобится для дальней привязки github. Заодно не забудем зарегистрироваться на github (у меня уже была регистрация, но в данном случае это почти ничего не меняет, так как делаю я с новой виртуальной машины).

(**?@fig-004**)

(**?@fig-005**)

(**?@fig-006**)

(**?@fig-007**)

Теперь добавили ключ на Github, таким образом подключили систему к Github.

(**?@fig-008**)

Настроим автоматические подписи коммитов Git, заставим Git использовать мой email для подписи коммитов.

(**?@fig-009**)

Произведём авторизацию и создадим шаблон рабочего пространства, для этого создадим папку и в ней образуем репозиторий, скопируем репозиторий и, перейдя в каталог курса, удалим всё лишнее и создадим каталоги.

(**?@fig-010**)

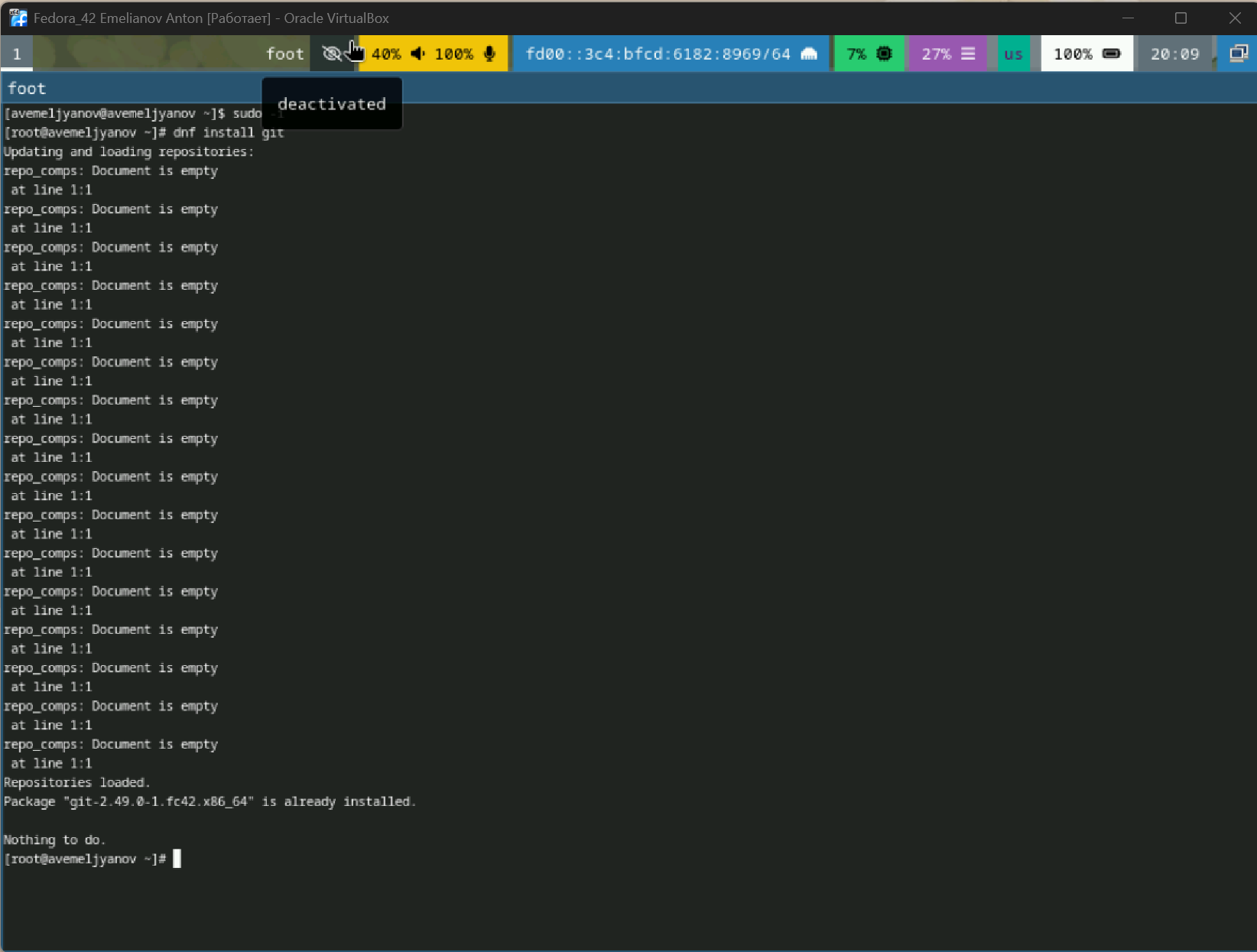
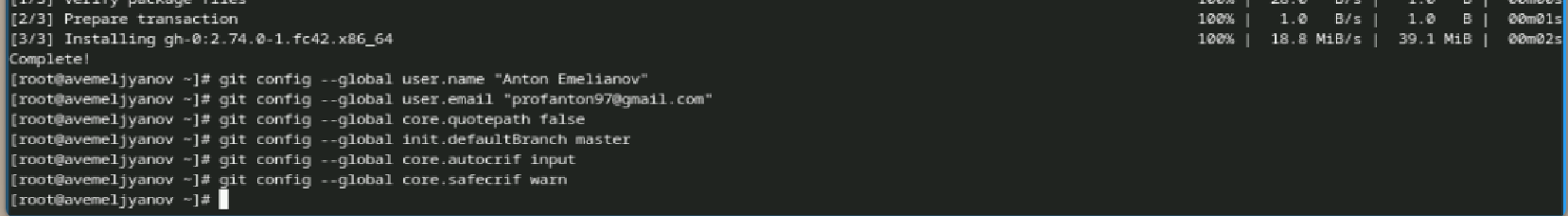
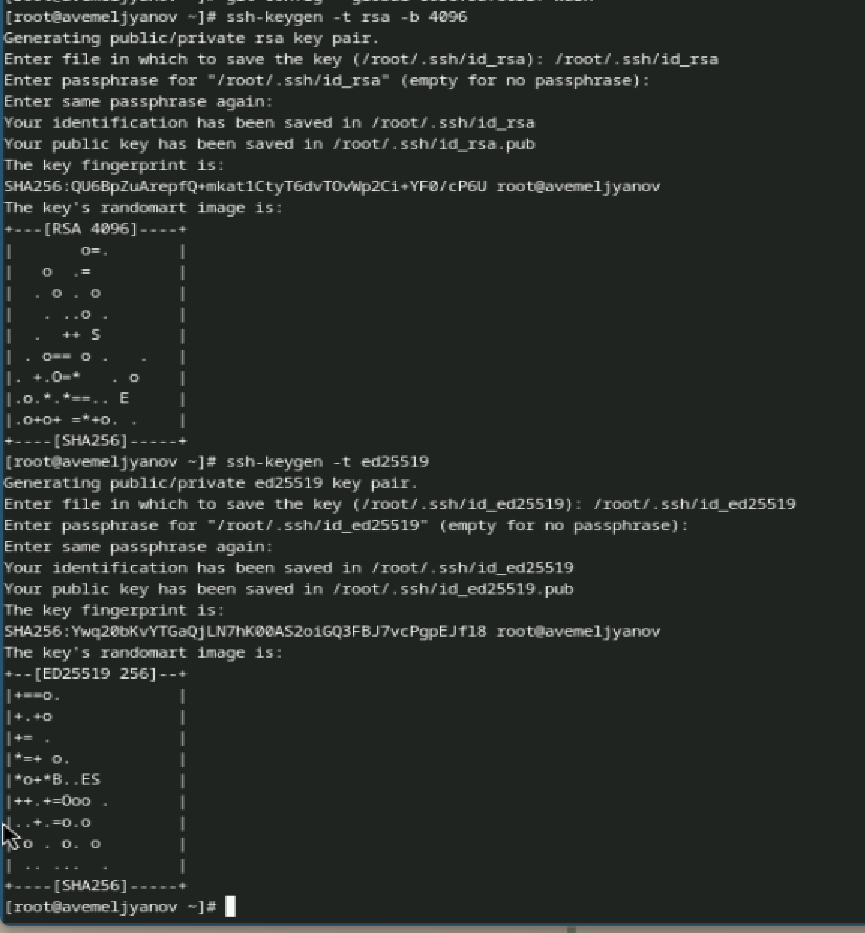
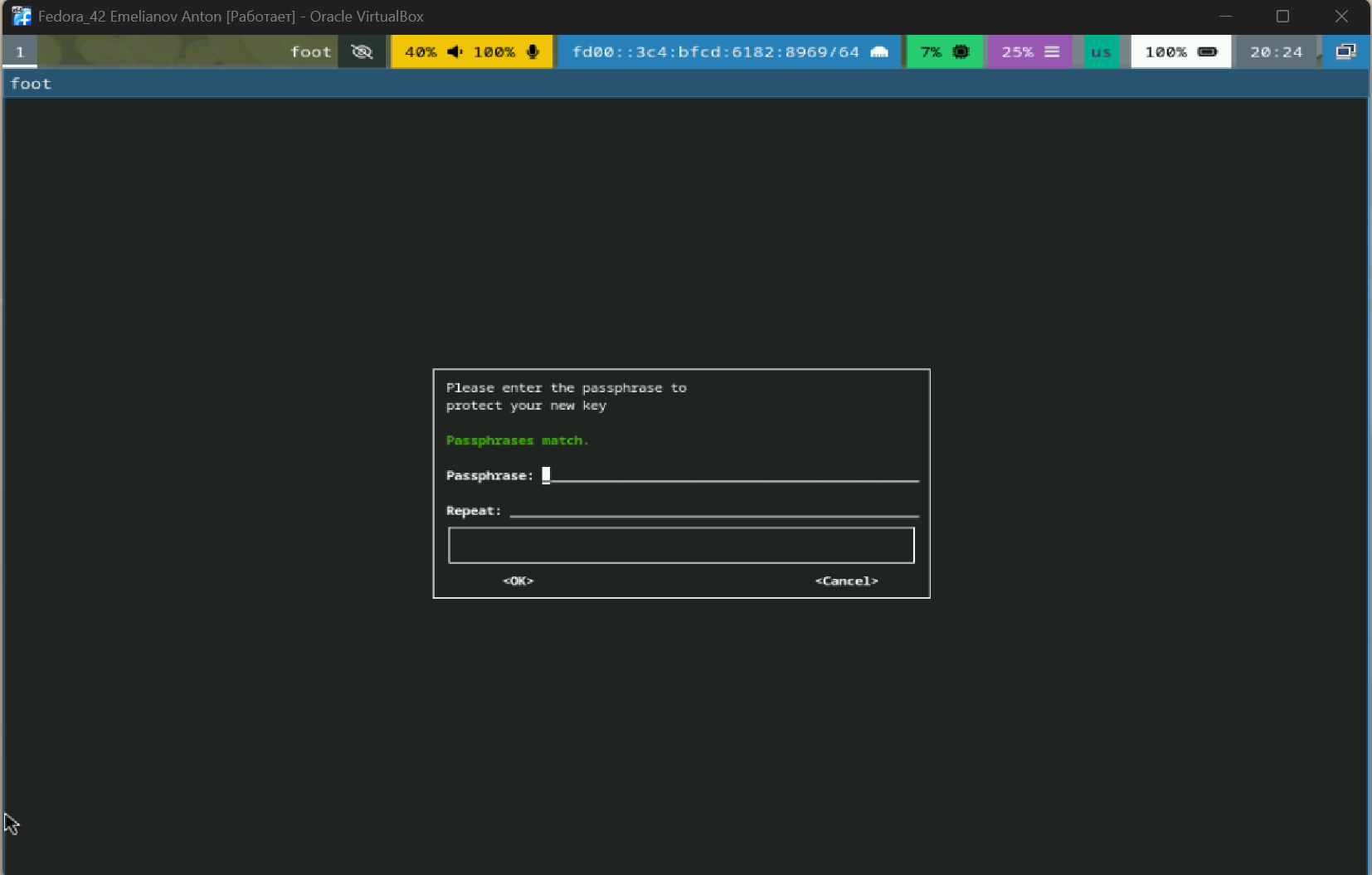
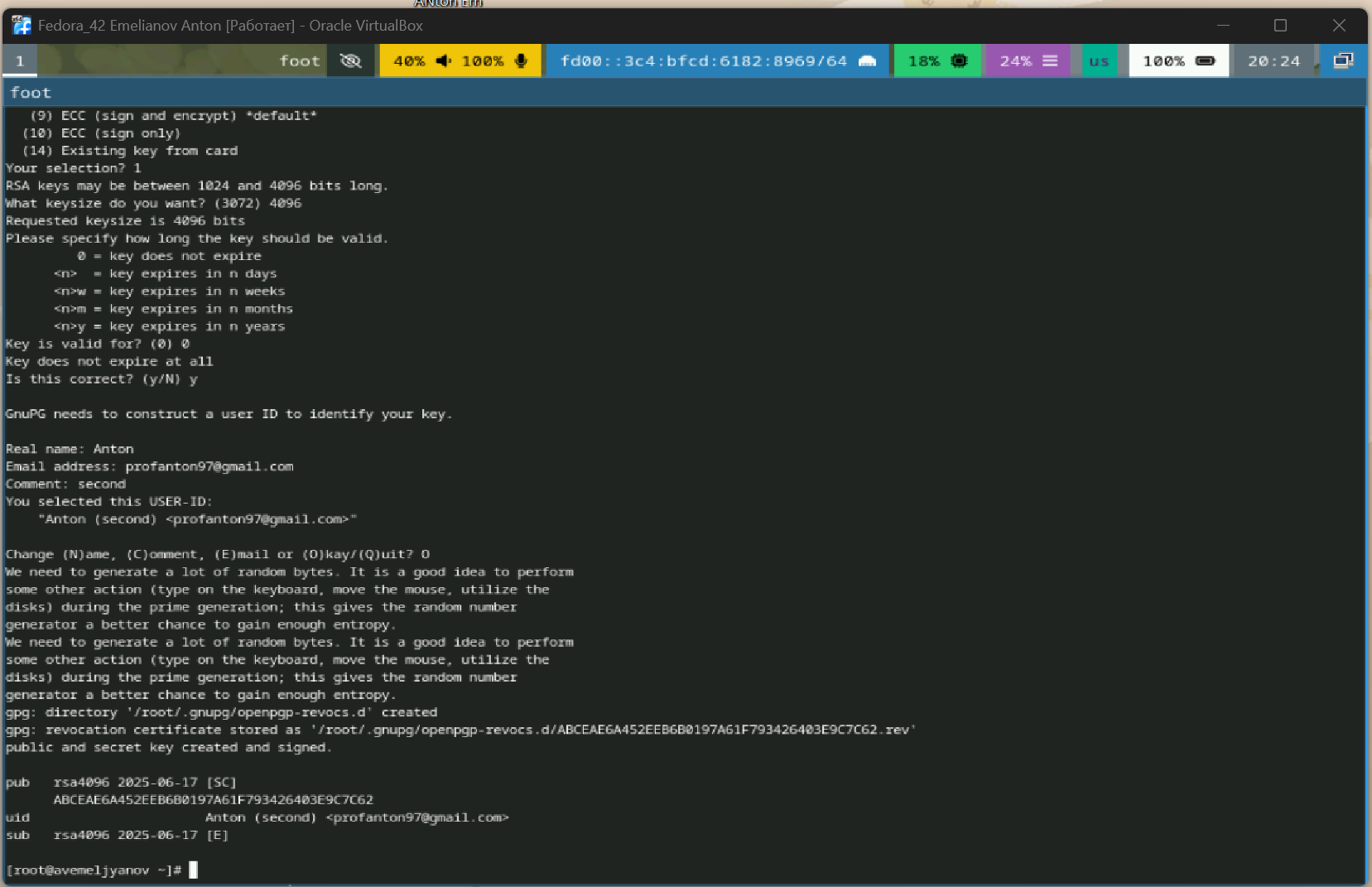
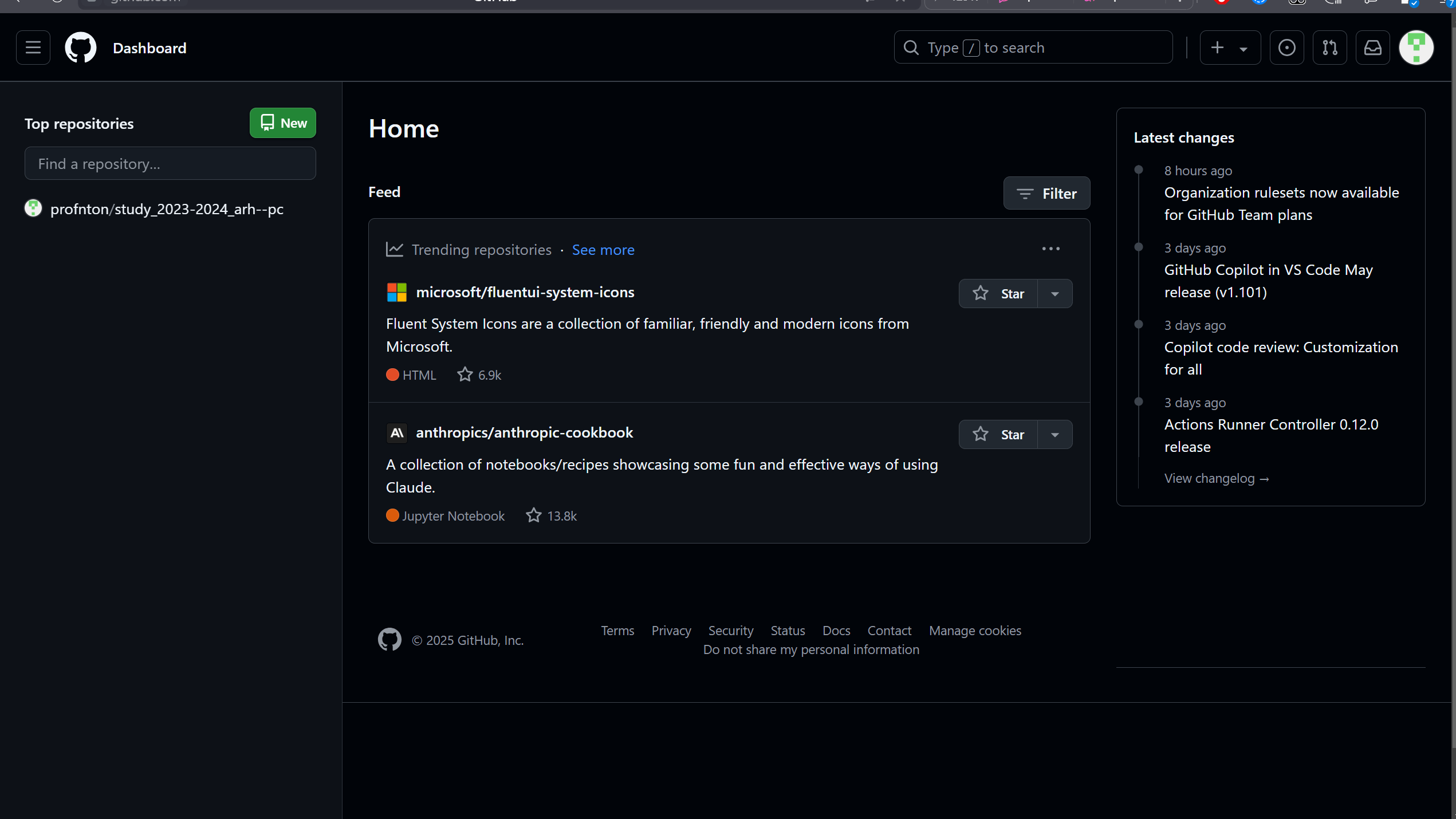
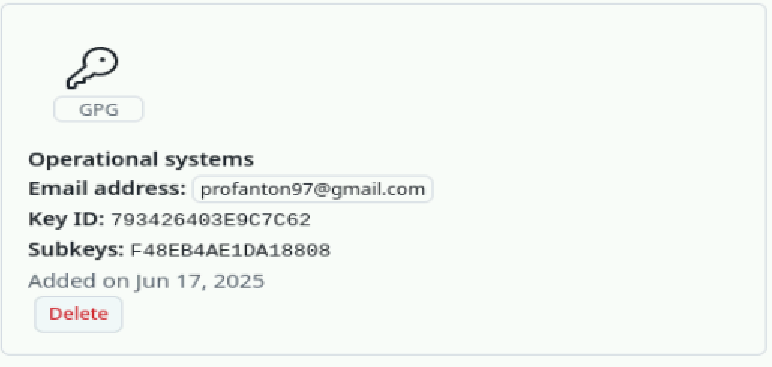
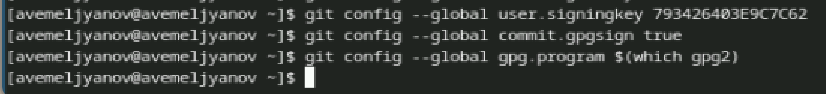
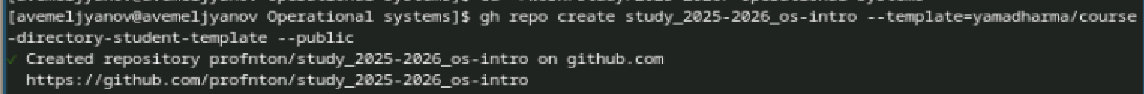
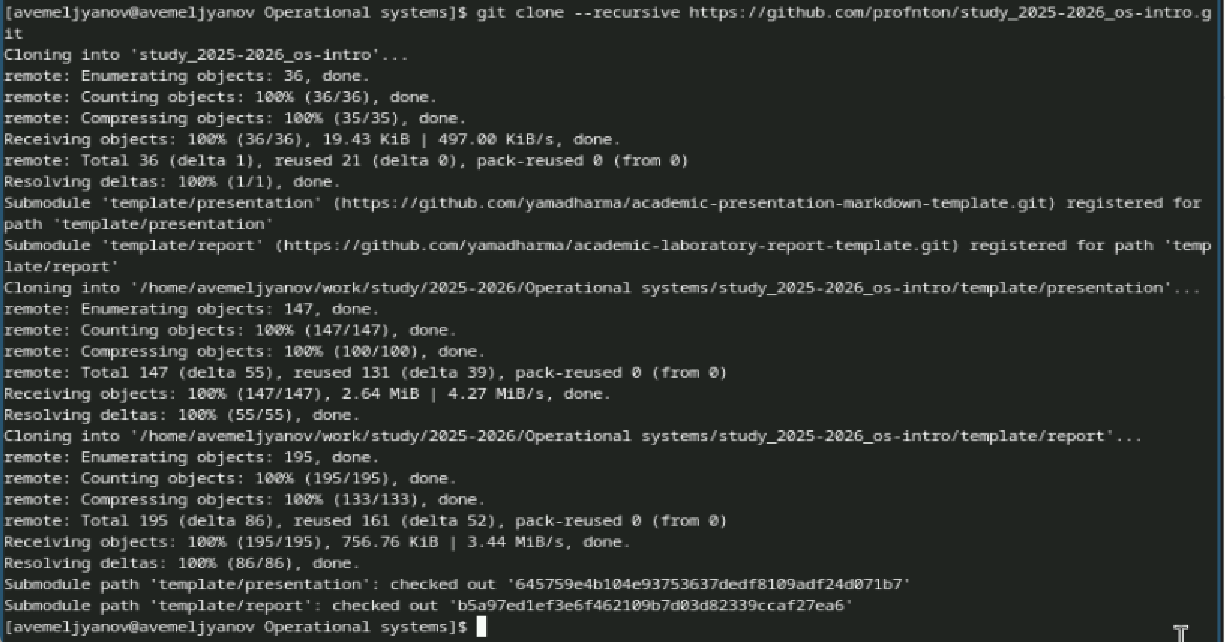
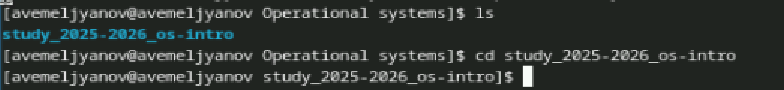
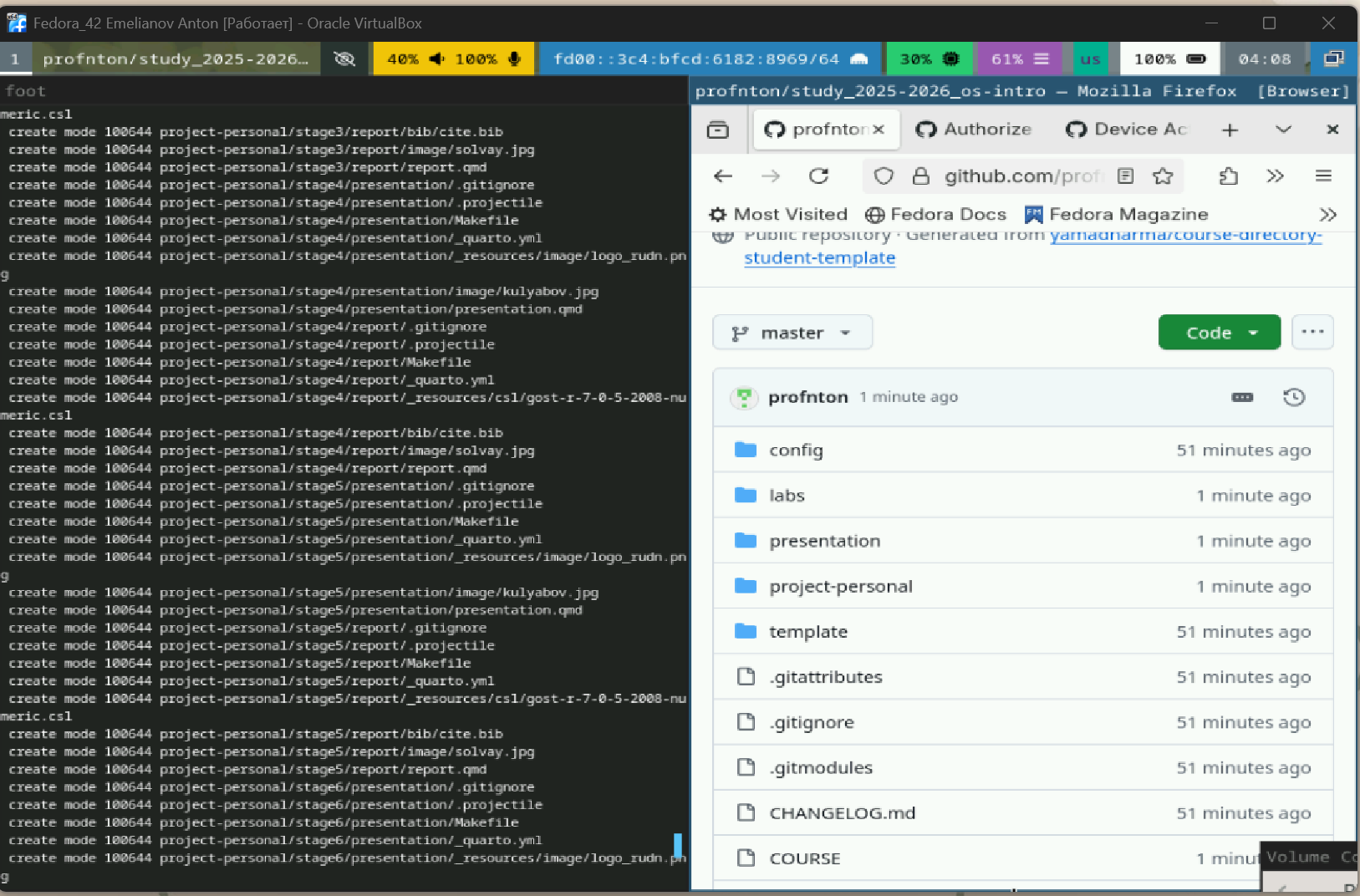
(**?@fig-011**)

(**?@fig-012**)

(**?@fig-013**)

Затем отправим файлы на сервер.

(**?@fig-014**)

 Установка gh прошла успешно.{#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} Подрубаем github.{#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%} {#fig-001 width =70%}

# 5. Выводы

В этой лабораторной работе мы успешно создали и настроили систему контроля версий Git, её окружение и организовали её взаимодействие с Github.

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? VCS используется для контроля изменения файлов во времени, их основная задача - контроль версий, сравнение, возврат, совместная работа, ветвление и резервное копирование Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище = все версии проекта, а также история их изменений Commit = снимок состояния проекта в момент времени История = последовательность Commit-ов

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.