Лабараторная работа 2

Простейший вариант

Хайманов Асланбек Султанович

Содержание

# Цель работы

Освоить умения по работе с git

# Задание

* Установка и настройка git
* Создание ключей
* Настройка каталога курса
* Контрольные вопросы

# Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. [-@tbl:std-dir] приведено краткое описание стандартных каталогов Unix.

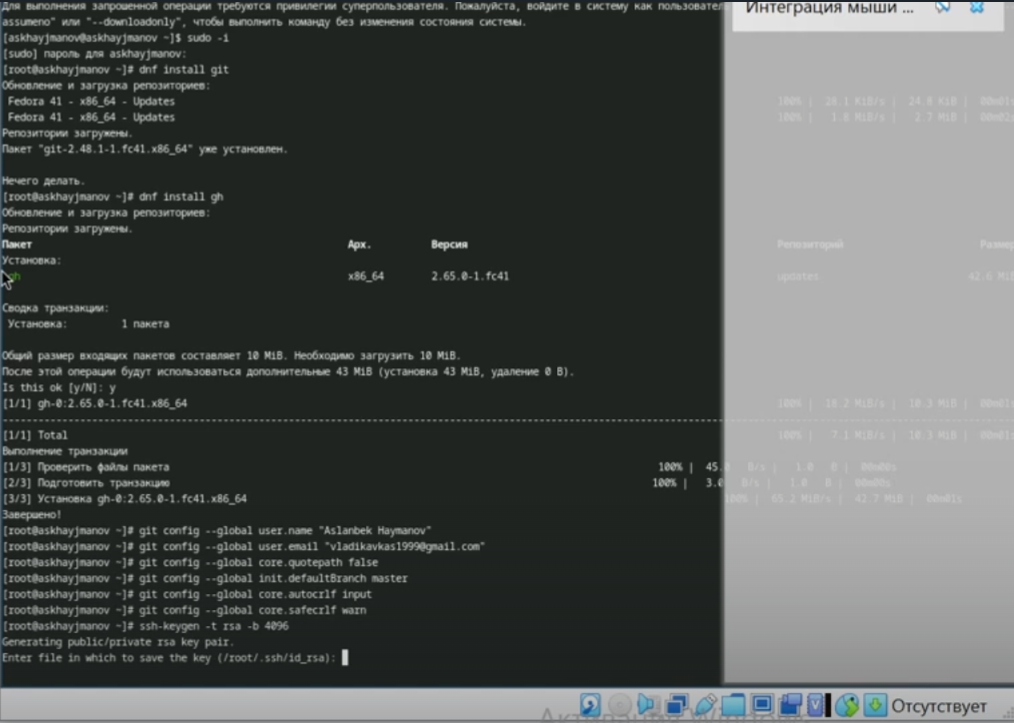
Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir}

| 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 [@tanenbaum\_book\_modern-os\_ru; @robbins\_book\_bash\_en; @zarrelli\_book\_mastering-bash\_en; @newham\_book\_learning-bash\_en].

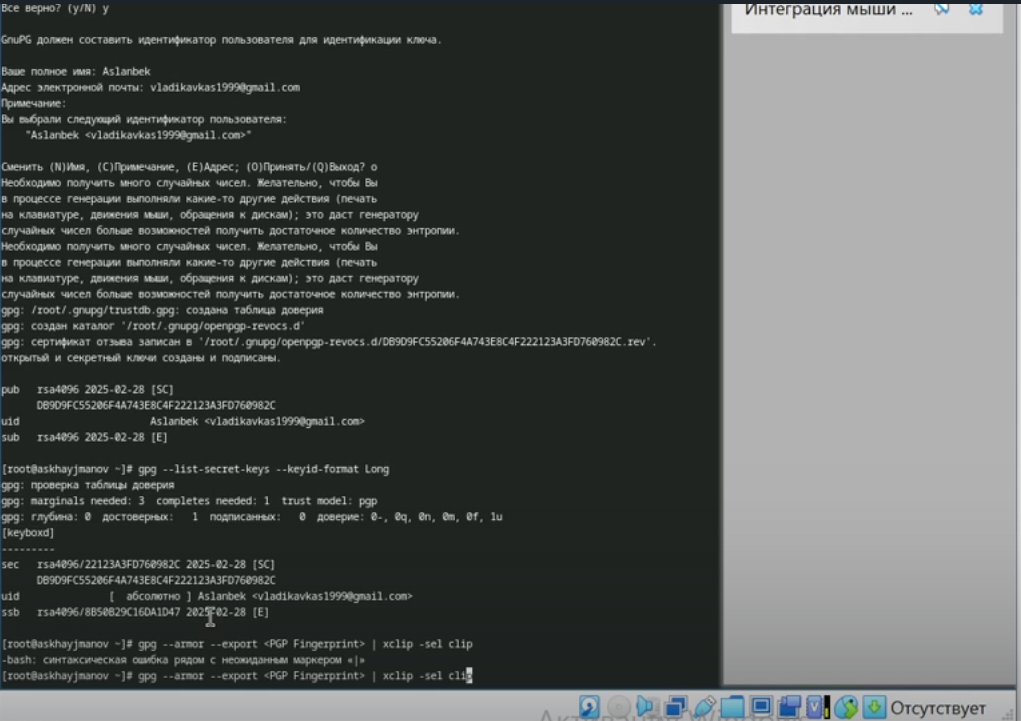
# Выполнение лабораторной работы

## Установка и настройка git



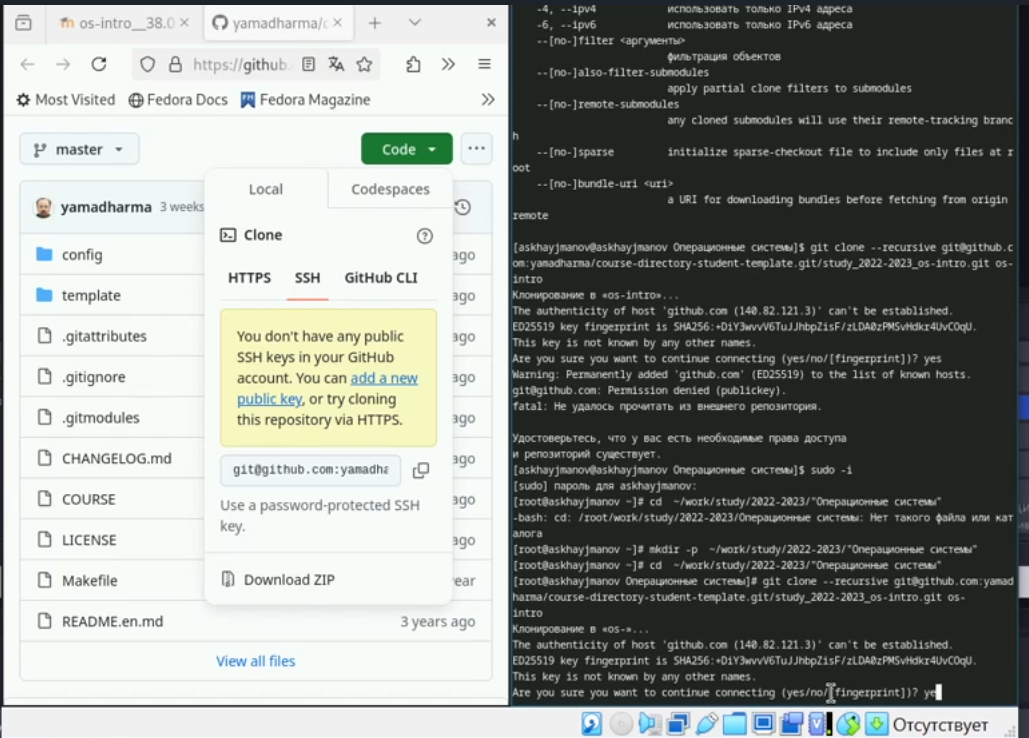
Установка и настройка гит (рис 1)

## Создание ключей



Создание ключей (рис 2)

## Настройка каталога курса



Настройка каталога (рис 3)

## Контрольные вопросы

* Системы контроля версий (VCS) — это инструменты для отслеживания изменений в файлах и управления проектами. Они позволяют сохранять различные версии файлов, что упрощает возврат к предыдущим состояниям и совместную работу.
* Основные понятия VCS
  + Хранилище (Repository): Место, где хранятся файлы проекта и их история изменений.
  + Commit: Операция, сохраняющая изменения в хранилище с описанием внесённых изменений.
  + История (History): Последовательность всех коммитов, показывающая, как проект развивался.
  + Рабочая копия (Working Copy): Локальная версия файлов, с которой пользователь работает.
* Централизованные и децентрализованные VCS
  + Централизованные VCS: Один центральный сервер, к которому обращаются все пользователи. Примеры: Subversion (SVN), CVS.
  + Децентрализованные VCS: Каждый пользователь имеет полную копию репозитория. Примеры: Git, Mercurial.
* Действия с VCS при единоличной работе
  + Создание локального репозитория.
  + Внесение изменений в рабочую копию.
  + Сохранение изменений с помощью commit.
  + Просмотр истории изменений.
  + Возврат к предыдущим версиям.
* Порядок работы с общим хранилищем VCS
  + Клонирование удалённого репозитория.
  + Создание новой ветки.
  + Внесение изменений и commit.
  + Синхронизация изменений (push).
  + Получение изменений от других (pull).
* Основные задачи Git
  + Отслеживание изменений.
  + Управление версиями.
  + Слияние изменений.
  + Работа с ветками.
  + Возврат к предыдущим версиям.
* Команды Git
  + git init: Создаёт новый репозиторий.
  + git clone: Клонирует удалённый репозиторий.
  + git add: Добавляет изменения в индекс.
  + git commit: Сохраняет изменения.
  + git push: Отправляет изменения в удалённый репозиторий.
  + git pull: Получает изменения из удалённого репозитория.
  + git branch: Управляет ветками.
* Примеры использования
  + Локальный репозиторий:git init git add . git commit -m “Initial commit”
  + Удалённый репозиторий:git clone https://github.com/user/repo.git git push origin main
* Ветви позволяют создавать параллельные линии разработки, что полезно для работы над новыми функциями или исправлениями, не затрагивая основную ветку.
* Игнорирование файлов при commit
* Файл .gitignore используется для указания файлов и папок, которые Git должен игнорировать, чтобы избежать случайного коммита ненужных файлов. Это помогает поддерживать чистоту репозитория.

# Выводы

Здесь кратко описываются итоги проделанной работы.

# Список литературы