МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 5**

по дисциплине:” Системное программирование”

на тему: “ Администрирование системы Linux”

Выполнил**:** студент группы 10702121

Меркулова М. С.

Принял**:** Давыденко Н.В.

Минск 2023

# Лабораторная работа № 5. Администрирование системы Linux

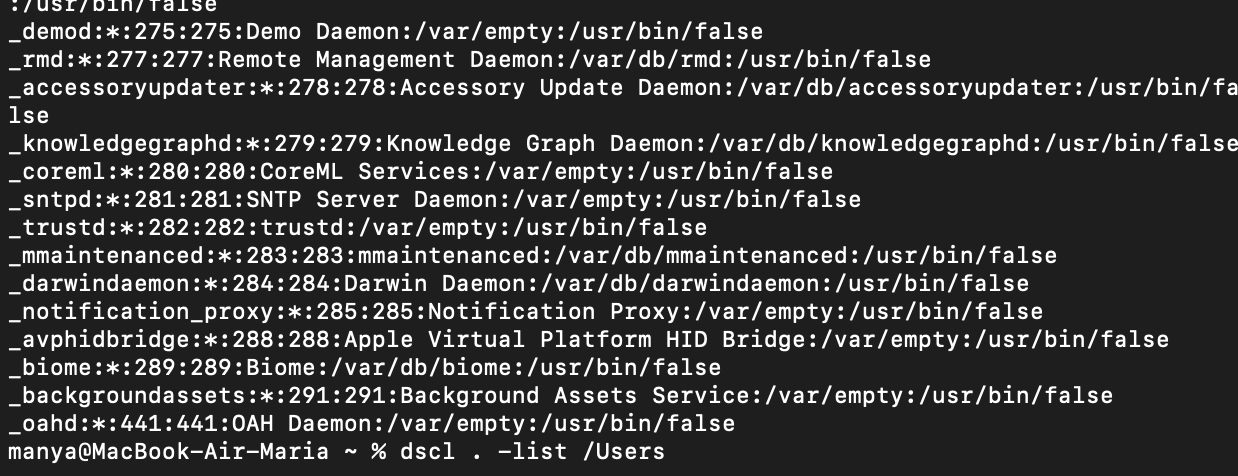
Цель работы: Закрепить на практике основы администрирования системы Linux, изучить атрибуты файлов и права доступа к ним, освоить работу с файлами и каталогами.

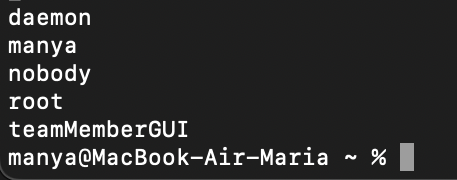
**Задание 1**

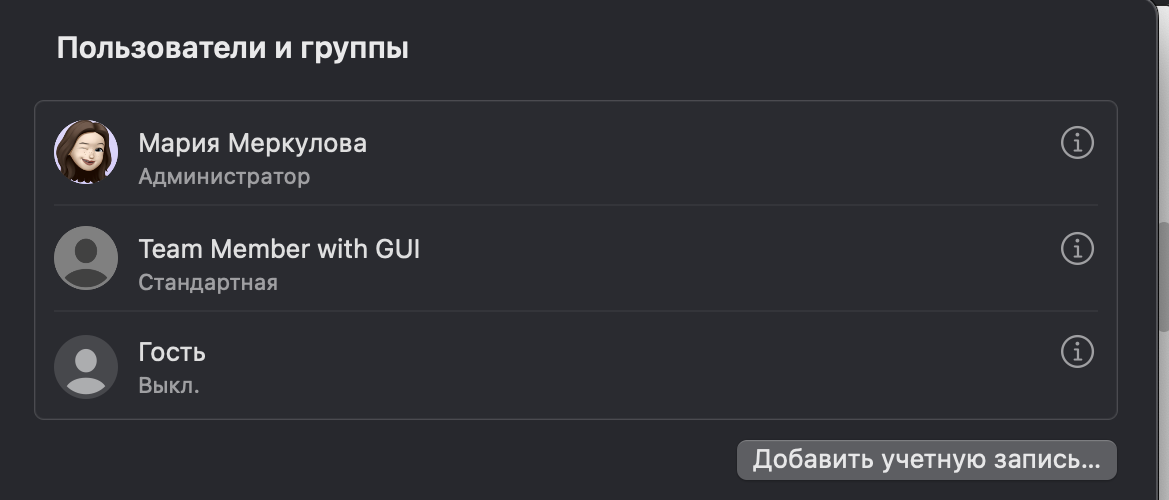
Создайте *стандартного пользователя* с помощью GUI. Пользователю присвойте имя члена вашей команды, но в имени должен присутствовать суфффикс \*GUI. Это нужно для проведения экспериментов. При присвоении имен действуйте по принципу: «Относитесь к именам переменных, как к именам детей своих». Пароль пользователя должен быть простым, например «123».



Просмотрите содержимое файла **/etc/passwd**, сравните атрибуты *реальных пользователей* и пользователя **root**. Результат поддтвердите скриншотом. Дайте пояснения.







Пояснение:  **/etc/passwd файл**: Файл **/etc/passwd** в Unix-подобных операционных системах (включая macOS) содержит информацию о пользователях на системе. Каждая строка в этом файле представляет одного пользователя и включает различные атрибуты, такие как имя пользователя (Username), уникальный идентификатор пользователя (UID), группа пользователя (GID), полное имя, каталог пользователя и оболочка (shell), которую пользователь использует при входе в систему.

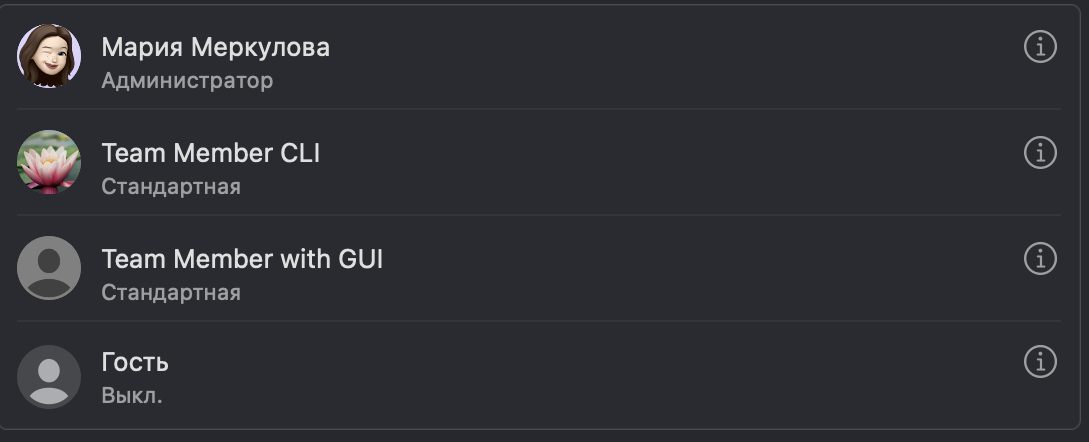
**Сравнение атрибутов пользователя root и других пользователей**:

• Пользователь **root** обычно имеет UID 0, что означает, что он имеет абсолютные административные права.

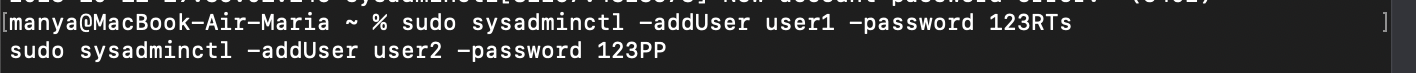
• У пользователей обычно UID больше 0.

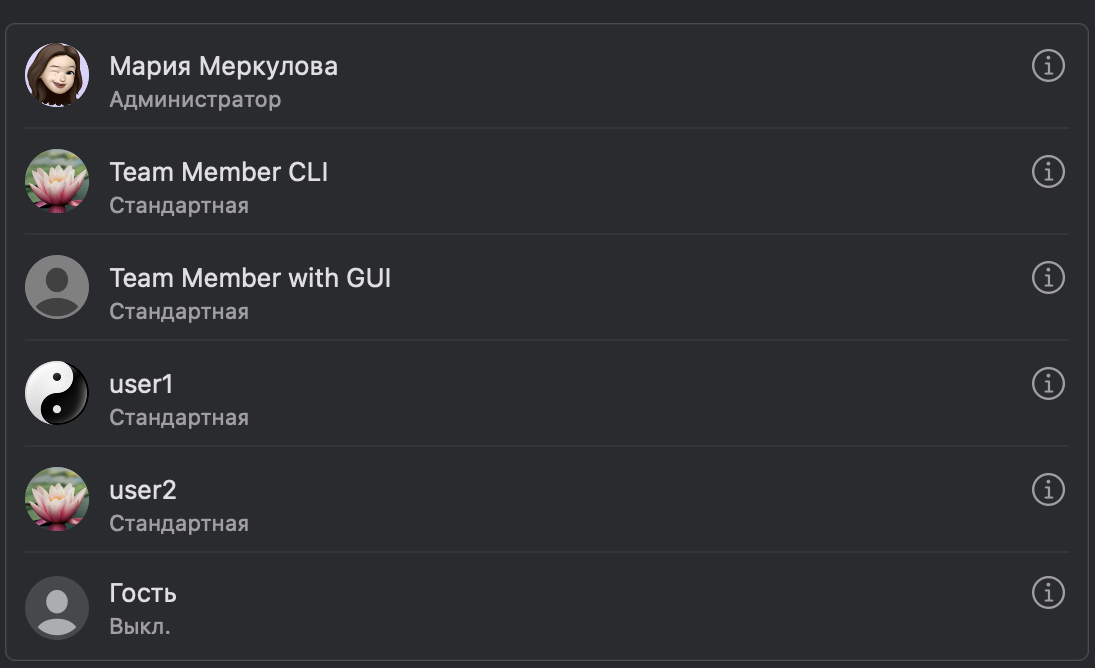
• Оболочка **root** часто установлена как **/bin/sh**, что представляет командную оболочку Shell. В то время как у других пользователей, оболочка может быть установлена как **/bin/bash** или другая командная оболочка.

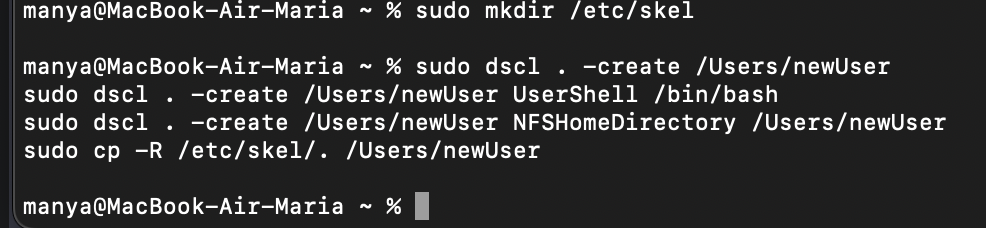
Создайте несколько *стандартных пользователей* (аккаунтов) посредсвам *командной строки*. При этом пользователям присвойте имена членов вашей команды, но в имени должен присутствовать суфффикс \*CL или \*CLI. Это нужно для того, чтобы отличить пользователей созданных при помощи графического и консольного интерфейсов.

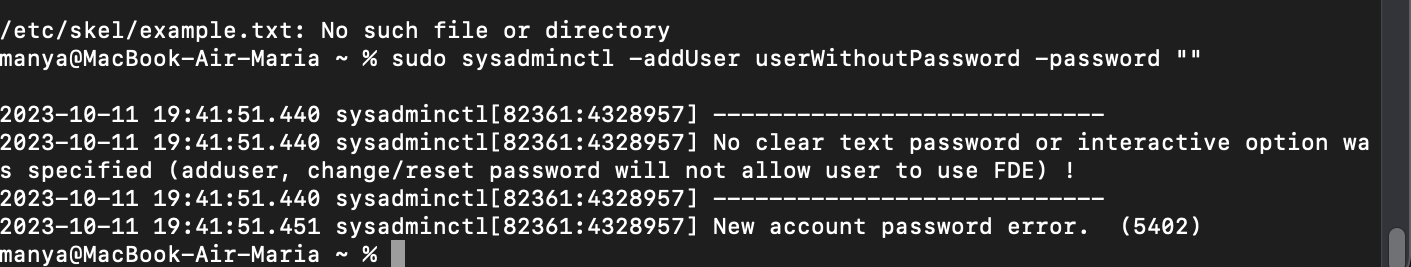


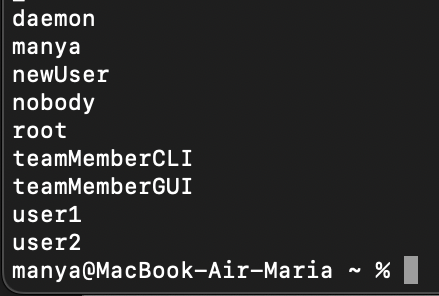
Создайте двуъ пользователей с одинаковыми простыми паролями. Создайте *скелет* и *пользователя* с *шаблоном скилета* в директории /home. Также попробуйте создать аккаунт реального пользователя без пороля.



****

****

****

****

**Задание 2**

Пароли должны быть простые и **одинаковые**. Это нужно для проведения экспериментов. Просмотрите содержимое файла **/etc/passwd**, сравните атрибуты *реальных пользователей*. Просмотрите содержимое файла **/etc/shadow**, сравните атрибуты паролей пользователей, особое внимание обратите на шифр пароля у пользователей с однаковым паролем.

Измените периоды изменения паролей для пользователей. Внесите ограничения в следующие атрибуты: минимальный возраст пароля (сутки); максимальный возраст пароля (сутки); период предупреждения пароля; пе- риод бездействия пароля; дата истечения срока действия аккаунта.

Результат поддтвердите скриншотом. Дайте пояснения.





Затем мы используем команду

minAge 1

maxAge 90

Перегружаем систему и получаем результат, в результате которого пароль нужно обновлять или можно обновлять от 1 до 90 дней

Так же в системе можно сделать

**• minChars** - минимальное количество символов в пароле.

**• minLowerCase** - минимальное количество символов в нижнем регистре.

**• minUpperCase** - минимальное количество символов в верхнем регистре.

**• minDigits** - минимальное количество цифр.

**• minSpecial** - минимальное количество специальных символов.

**• minAge** - минимальный возраст пароля (сутки).

**• maxAge** - максимальный возраст пароля (сутки).

**• warnPeriod** - период предупреждения (сутки).

**• inactivePeriod** - период бездействия пароля (сутки).

**• accountExpiresAfter** - дата истечения срока действия аккаунта (гггг-мм-дд)

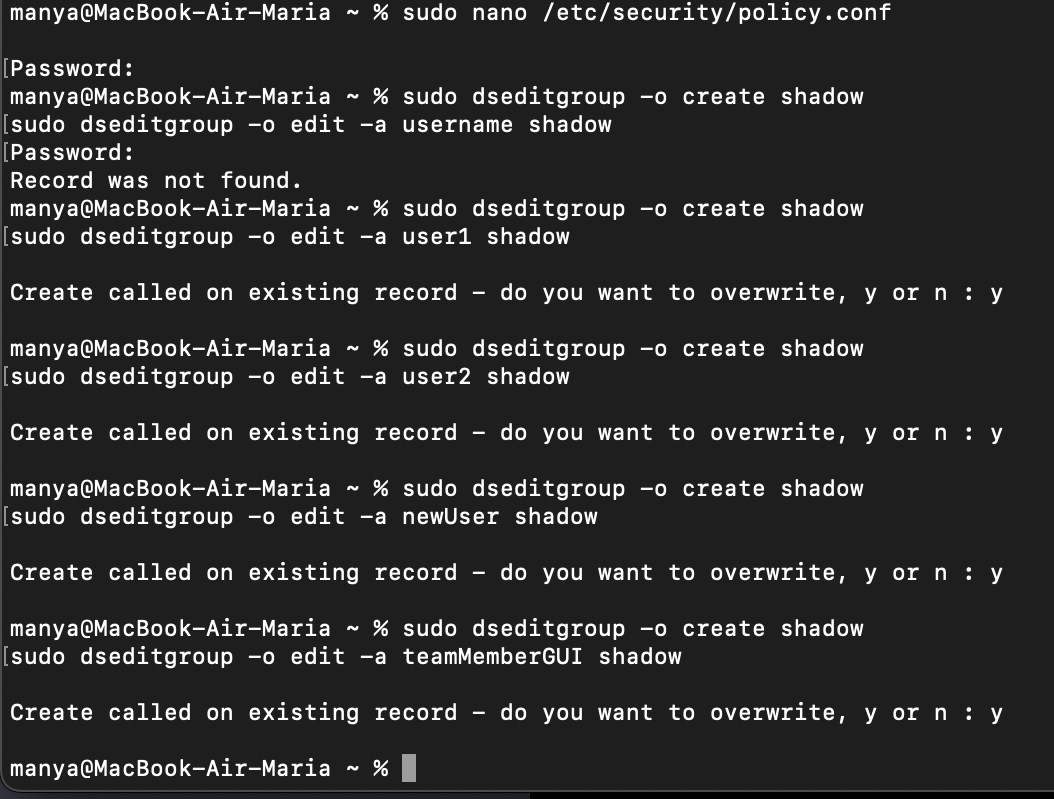
**Решение**

Создайте пользователя (аккаунт) с правами администратора. В каталогах /home/ администраторов и других пользователей создайте по несколько файлов с разными правами доступа.

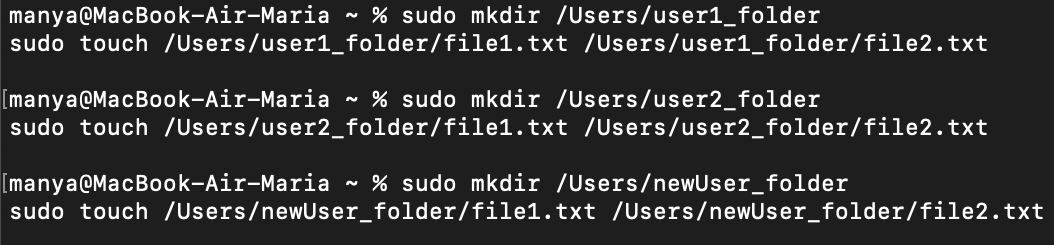
**В macOS учетные записи с правами администратора создаются через системные настройки, и создание таких аккаунтов через командную строку непосредственно не поддерживается.**

Создайте общую группу пользователей. Выполните следующие требования:

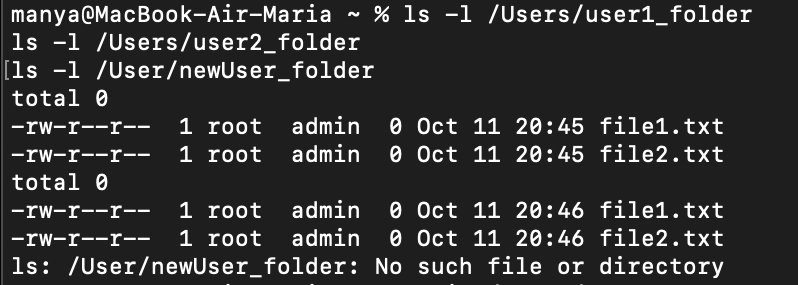
1. Одного из пользователей перевести в группу shadow



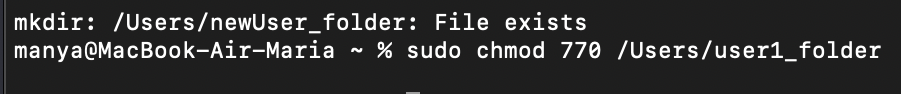
1. Создать у каждого пользователей директорию с 2 файлами



1. Просмотреть текущие права доступа к файлам для всех пользователей



1. Каталог пользователя в группе shadow сделать доступным только в своей группе

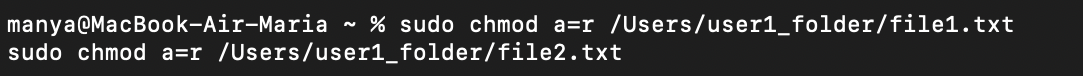


1. Файлы второго пользователя сделать доступными только владельцам



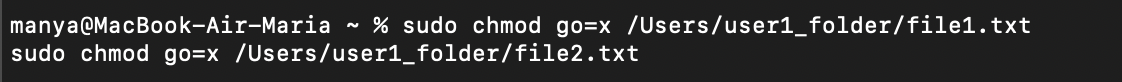
1. Под админом назначить всем созданным файлам права только для чтения для всех

пользователей

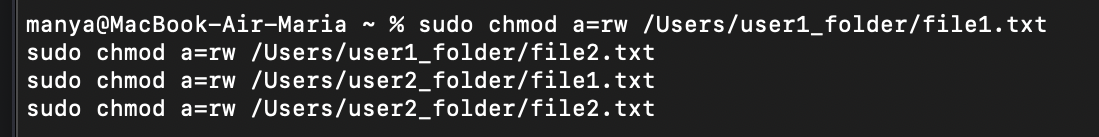


1. Пользователем в группе shadow лишить всех остальных пользователей права ис-

полнять его файлы



1. Под админом назначить всем пользователям все права

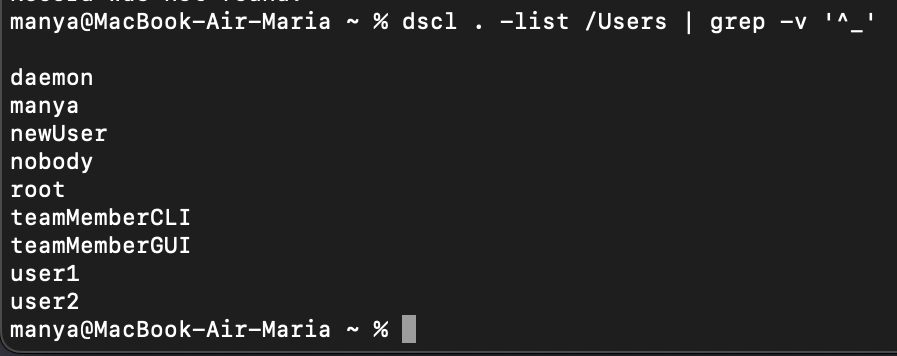
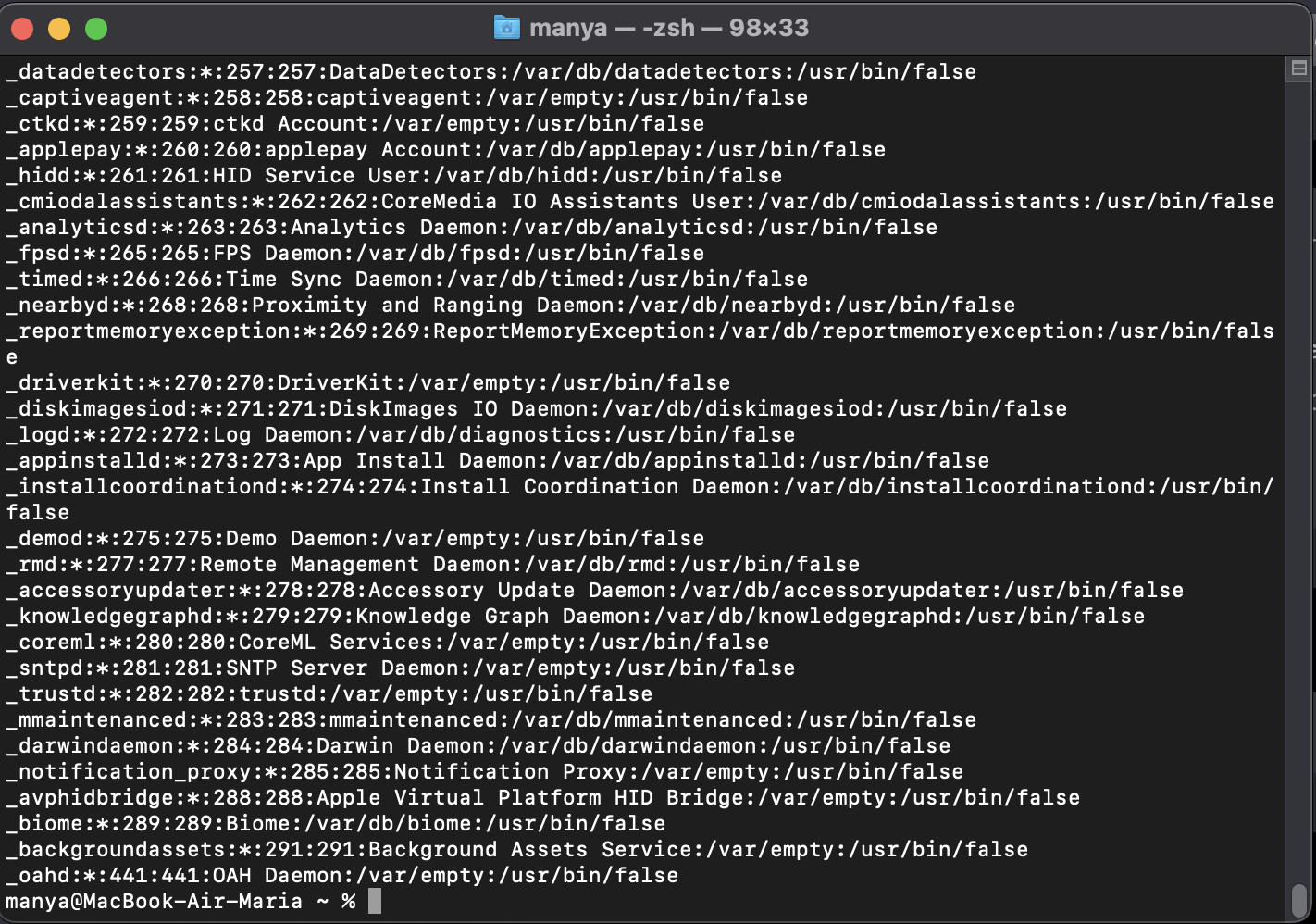


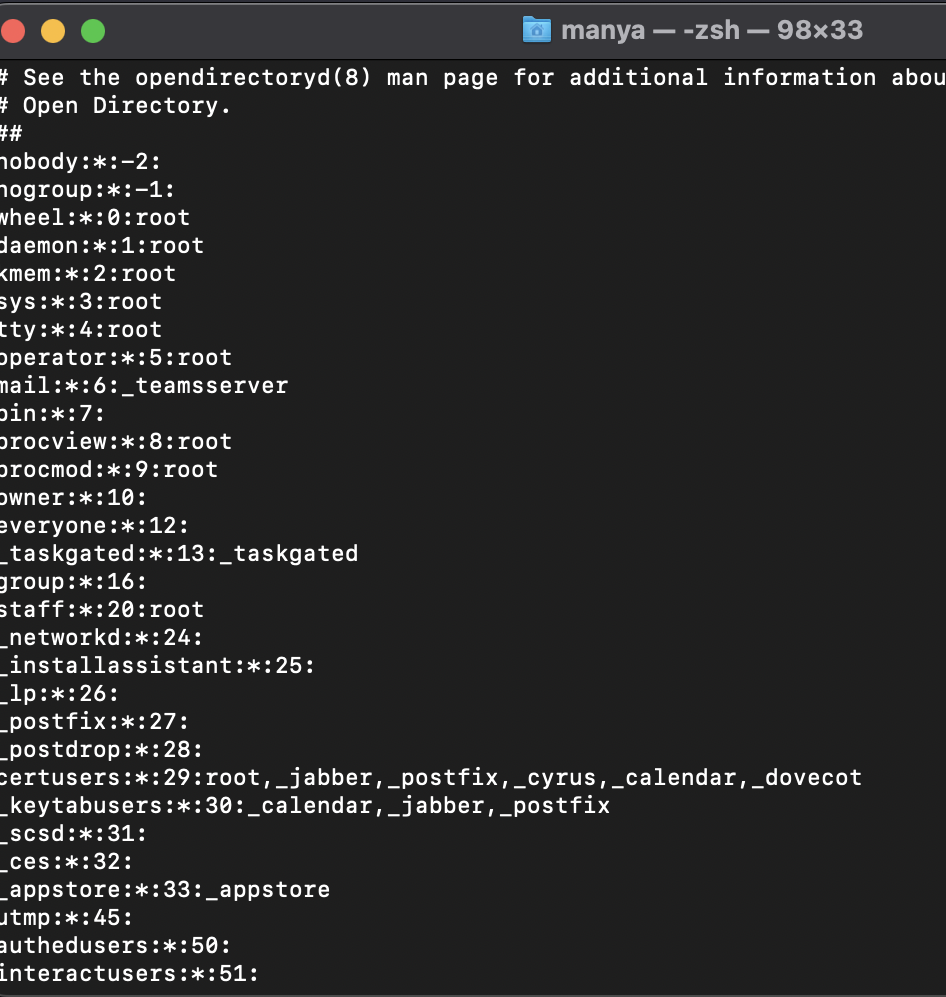
1. Удалить пользователя, находящегося в группе shadow



Укажите является ли группа системной или создана пользвателем.

Получите индентификаторы пользователей и состояние активных пользо- вателей системы. Проверьте содержимое файлов /etc/shadow, /etc/passwd, /etc/group.

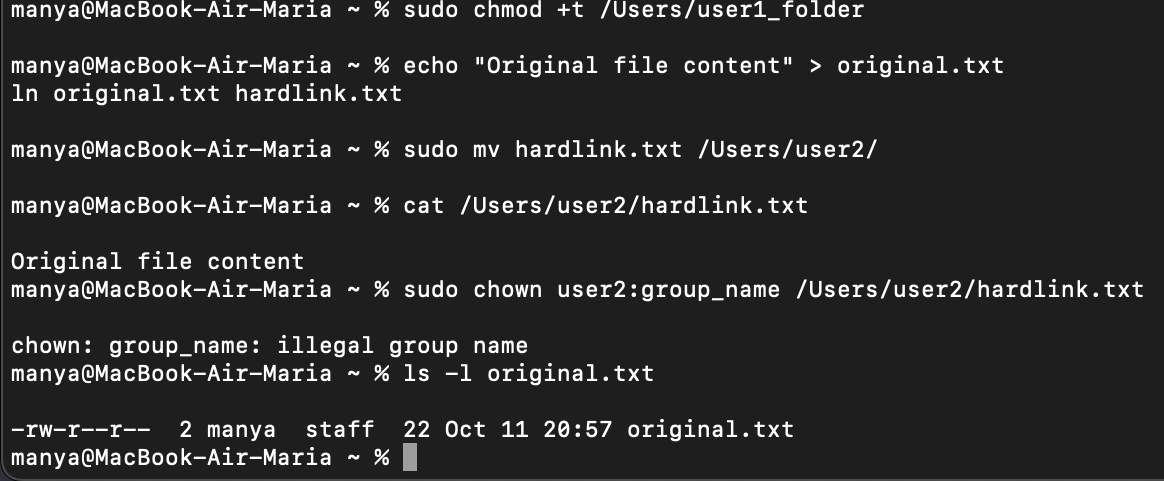


Некоторые файлы защитите *липким битом*. Попробуйте удалить созданные файлы из других аккаунтов администраторов и простых пользователей. Опишите какой получили результат.



Если попытаться удалить файл, который не принадлежит текущему пользователю и не имеет для него прав на запись, то будет выдана ошибка.

Создайте *жесткую ссылку*, перенесите эту ссылку в пространство другого пользователя и откройте ее в сеансе этого пользователя, затем присвойте жесткой ссылке одну группу и откройте ссылку из пользователя этой группы. Измените владельца и группу жесткой ссылки и посмотрите, как изменились атрибуты основного файла.



После выполнения этих шагов, можно увидеть, что атрибуты **original.txt** изменились в соответствии с группой и владельцем жесткой ссылки **hardlink.txt**.

Контрольные вопросы:

1. Какая концепция прав доступа к файлу реализована в ядре?

В ядре Unix-подобных операционных систем (включая macOS, Linux и другие), реализована концепция прав доступа к файлам с использованием системы управления правами доступа (Access Control Lists, ACL) и масок прав доступа.

Концепция прав доступа основана на следующих основных элементах:

**Пользователи и группы**: Каждый пользователь и группа имеют свой уникальный идентификатор, называемый UID (User Identifier) для пользователей и GID (Group Identifier) для групп.

**Права доступа**: Каждый файл или каталог имеет ассоциированные права доступа, которые определяют, какие операции могут быть выполнены с этим файлом (чтение, запись, выполнение и другие).

**Владелец и группа файла**: Файл имеет владельца (пользователь, который создал файл) и группу, которая определяет группу пользователей, имеющих доступ к этому файлу.

**Маска прав доступа (umask)**: Маска прав доступа определяет значения по умолчанию для прав доступа при создании нового файла или каталога. Она вычитается из максимального набора прав доступа (обычно 777 для файлов и 777 для каталогов), чтобы установить права доступа по умолчанию.

**Access Control Lists (ACLs)**: ACLs предоставляют дополнительные детализированные права доступа к файлам и каталогам. Они позволяют задавать права для конкретных пользователей и групп вне обычных прав доступа.

**Липкий бит (sticky bit)**: Липкий бит устанавливается на каталогах и предотвращает удаление файлов другими пользователями, даже если у них есть права на запись в этот каталог.

Эти концепции прав доступа реализованы в ядре операционной системы и используются для обеспечения безопасности файловой системы и управления доступом к файлам и каталогам. Они позволяют определить, кто может выполнять какие операции с файлами, обеспечивая конфиденциальность и целостность данных.