**Лабораторная работа 2.**

**Разграничение прав доступа**

**Серт Серкан Группа 8**

**Цель**: Изучение механизмов управления доступа к ресурсам, прав доступа. Постижение понятия пользователя и группы. Приобретение практических навыков управления пользователями при помощи консольных утилит. Приобретение навыков работы с правами пользователей и правами на файлы, каталоги при помощи консольных утилит.

1. **Какой uid у пользователя sit2? В какие группы он входит?**

**uid пользователя sit2 и группы, в которые он входит, можно узнать с помощью команды id. В данном случае, для пользователя sit2, uid будет равен 1001.Для того, чтобы узнать группы, в которые входит пользователь sit2, необходимо выполнить команду groups sit2**[**8**](https://rwsite.ru/gruppy-polzovatelej-v-linux/)**. Это покажет список групп, в которых пользователь sit2 является участником**

1. **Почему попытка удалить пользователя не удалась, и что нужно сделать для его удаления?**

Если попытка удалить пользователя не удалась, возможно, есть несколько причин:

* Пользователь выполняет какие-то процессы или активен: Если пользователь sit2 выполняет какие-то процессы или имеет открытые сеансы (например, сеанс входа в систему), то система не позволит удалить его, чтобы избежать возможных проблем с процессами.
* Файлы владеются пользователем: Если у пользователя sit2 есть файлы или директории, владельцем которых он является, то их также необходимо удалить или переназначить другому пользователю, прежде чем удалить учетную запись.
* Системный процесс зависит от пользователя: Иногда удаление пользователя может быть затруднено из-за зависимостей системных процессов или служб, использующих этого пользователя.

1.Убедитесь, что пользователь не активен

Убедитесь, что пользователь sit2 не выполняет никаких процессов и не имеет активных сеансов. Это можно проверить с помощью команды who или w:

2. Остановите процессы пользователя

Если вы видите активные сеансы пользователя, завершите их. Например, можно использовать **kill** для завершения процессов.

3. Измените владельца файлов

Перед удалением пользователя убедитесь, что файлы, принадлежащие пользователю **sit2**, переназначены другому пользователю или удалены.

4. Удалите пользователя

Используйте команду **userdel** для удаления пользователя **sit2**:

sudo userdel -r sit2

Опция **-r** удаляет также домашний каталог пользователя и все файлы в нем. Убедитесь, что перед удалением вы выполнили все необходимые шаги для предотвращения возможных проблем.

1. **Какие права доступа установлены на домашний каталог пользователя «sit» ?**

Права доступа состоят из трех троек символов. Первая тройка представляет права владельца файла, вторая представляет права группы файла и третья права всех остальных пользователей.

В нашем случае это :

«rwx» - Права владельца файла

«r-x» - Права группы файла

«r-x» - Права всех остальных на файл.

Символ «r» означает, что чтение (просмотр данных содержащихся в файле) разрешено, «w» означает запись (изменение, а также удаление данных) разрешено и «x» означает исполнение (запуск программы разрешен).

1. **Как рекурсивно изменить права доступа на файлы в каталоге?**

Команда chmod использует два и более аргументов: метод, описывающий как именно необходимо изменить права доступа с последующим именем файла или списком файлов, к которым необходимо применить эти изменения:

chmod +x /home/sit/itmo.sh

В примере выше в качестве метода указано +x. Как можно догадаться, метод +x указывает chmod, что файл необходимо сделать исполняемым для пользователя, группы и для всех остальных. Если мы решим отнять все права на исполнение файла, то сделаем вот так:

chmod +x /home/sit/itmo.sh

1. **Как можно осуществлять переключение между пользователями в рамках одного терминала?**

Для переключения между пользователями в рамках одного терминала в Linux можно использовать команду **su** (Switch User) или **sudo** (Superuser Do). Обе команды позволяют войти в систему под другим пользователем, но есть некоторые различия в их использовании и настройках.

1. **Как удалить пользователя при этом сохранив его домашний каталог и данные внутри него?**

Прежде чем удалять пользователя, рекомендуется создать резервную копию его домашнего каталога. Это обеспечит безопасность ваших данных, на случай если что-то пойдет не так. Например, вы можете скопировать содержимое домашнего каталога пользователя **username** в новый каталог **username\_backup**:

sudo cp -r /home/username /home/username\_backup

**Удаление пользователя**

Теперь можно удалить пользователя с сохранением его домашнего каталога. Для этого используйте команду **userdel** с опцией **-r**, которая удаляет пользователя вместе с его домашним каталогом:

sudo userdel -r username

1. **Какое значение umask нужно установить, чтобы владелец и группа имели право на чтение, запись и исполнение, а все остальные пользователи не имели никаких прав?**

В операционной системе Linux базовые права для каталога равны 0777 (rwxrwxrwx), а для файла 0666 (rw-rw-rw). Когда процесс создает новый файл, он указывает, какие права доступа нужно задать для данного файла. Зачастую запрашиваются права 0666 (чтение и запись всеми), что дает больше разрешений, чем необходимо в большинстве случаев. К счастью, каждый раз, когда в Linux создается новый файл, система обращается к параметру, называемому umask. Система использует значение umask чтобы понизить изначально задаваемые разрешения на что-то более разумное и безопасное. Вы можете просмотреть текущие настройки umask набрав umask в командной строке:

sit@ubuntu:~$

umask 0002

По умолчанию umask 0002 используется для обычного пользователя. С этой маской права по умолчанию, для каталога, равны 777–2=775, а для файла 666–2=664. Для суперпользователя (root) umask по умолчанию равна 0022. С этой маской права по умолчанию, для каталога, равны 777–22=755, а для файла 666–22=644. В Linux-системах значением по умолчанию для umask является 0022, что позволяет другим читать ваши новые файлы (если они могут до них добраться), но не изменять их. Чтобы автоматически обеспечивать больший уровень защищенности для создаваемых файлов, можно изменить настройки umask:

sit@ubuntu:~$

umask 0077

Такое значение umask приведет к тому, что группа и прочие не будут иметь совершенно никаких прав доступа для всех, вновь созданных файлов.

В отличие от «обычного» назначения прав доступа к файлу, umask задает какие права доступа должны быть отключены. Снова посмотрим на таблицу соответствия значений чисел и методов, приведенную выше:

Воспользовавшись этой таблицей видим, что последние три знака в 0077 обозначают – rwxrwx. umask показывает системе, какие права доступа отключить. Совместив первое и второе становится видно, что все права для группы и остальных пользователей будут отключены, в то время как права владельца останутся нетронутыми.

1. **Как рекурсивно снять все suid биты с файлов в каталоге?**

Если на исполняемый файл установлен бит suid, то при выполнении эта программа автоматически меняет «эффективный userID» на идентификатор того пользователя, который является владельцем этого файла. То есть, не зависимо от того - кто запускает эту программу, она при выполнении имеет права хозяина этого файла. Владельцем исполняемого файла passwd является пользователь

root: sit@ubuntu:~$ ls -l /usr/bin/passwd

-rwsr-xr-x 1 root wheel 17588 Feb 18 2016 /usr/bin/passwd

Обратите внимание, что вместо x в триплете прав доступа владельца стоит s. Это означает что для этой конкретной программы установлены биты suid и права на запуск.

1. **Как разрешить программе (файлу) исполняться?**

Чтобы разрешить программе или файлу исполняться в Linux, вы должны установить соответствующие разрешения на этот файл. Разрешение на выполнение обозначается битом прав доступа **x**. В зависимости от вашего доступа к файлу (владелец, группа, остальные пользователи), вы можете установить разрешение на выполнение с помощью команды **chmod** или **chown**.

1. **Что такое бит sticky? Для чего он предназначен?**

Пользователи могут переименовать или удалять файлы внутри каталога если у них есть права на запись в этом каталоге. Для каталогов, которыми пользуется большое количество пользователей, в особенности /tmp и /var/tmp, это вызывает много проблем, так как любой пользователь может писать, удалять и переименовывать любые файлы в них – даже если они им не принадлежат.

В Linux существует так называемый sticky bit. Когда для /tmp установлен sticky bit (командой chmod +t /tmp), единственные, кто могут удалить или переименовать файлы в /tmp – это суперпользователь, либо владельцы этих файлов и владельцы каталогов, в которых находятся файлы.

При использовании ls -ld, вы можете видеть sticky bit как t в позиции, где вы обычно видите разрешение на выполнение для других пользователей:

root@ubuntu:~$ ls -ld /data/account/

drwxr-sr-t. 2 root account 4096 Apr 30 21:28 /data/account/

1. **Зачем нужны uid и gid?**

UID (User ID) и GID (Group ID) - это числовые идентификаторы, используемые операционной системой для идентификации пользователей и групп. Они играют важную роль в управлении доступом к ресурсам системы, а также в разграничении прав доступа между различными пользователями и группами.

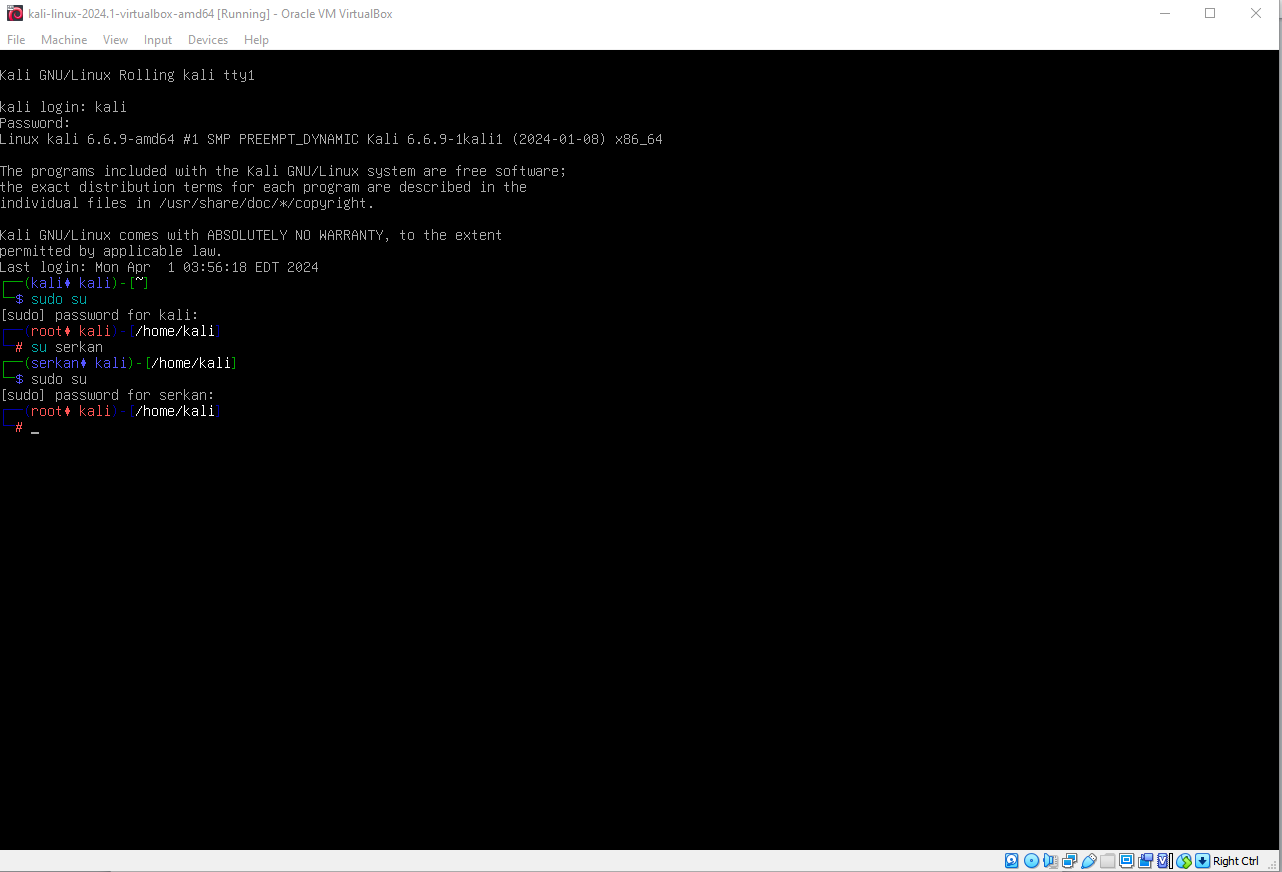
1. **Почему uid пользователя задается больше 1000?**

В операционной системе Linux обычно существует соглашение о том, что пользователи с UID (User ID) больше 1000 предназначены для обычных пользователей, в то время как пользователи с UID меньше 1000 зарезервированы для системных пользователей и служб.

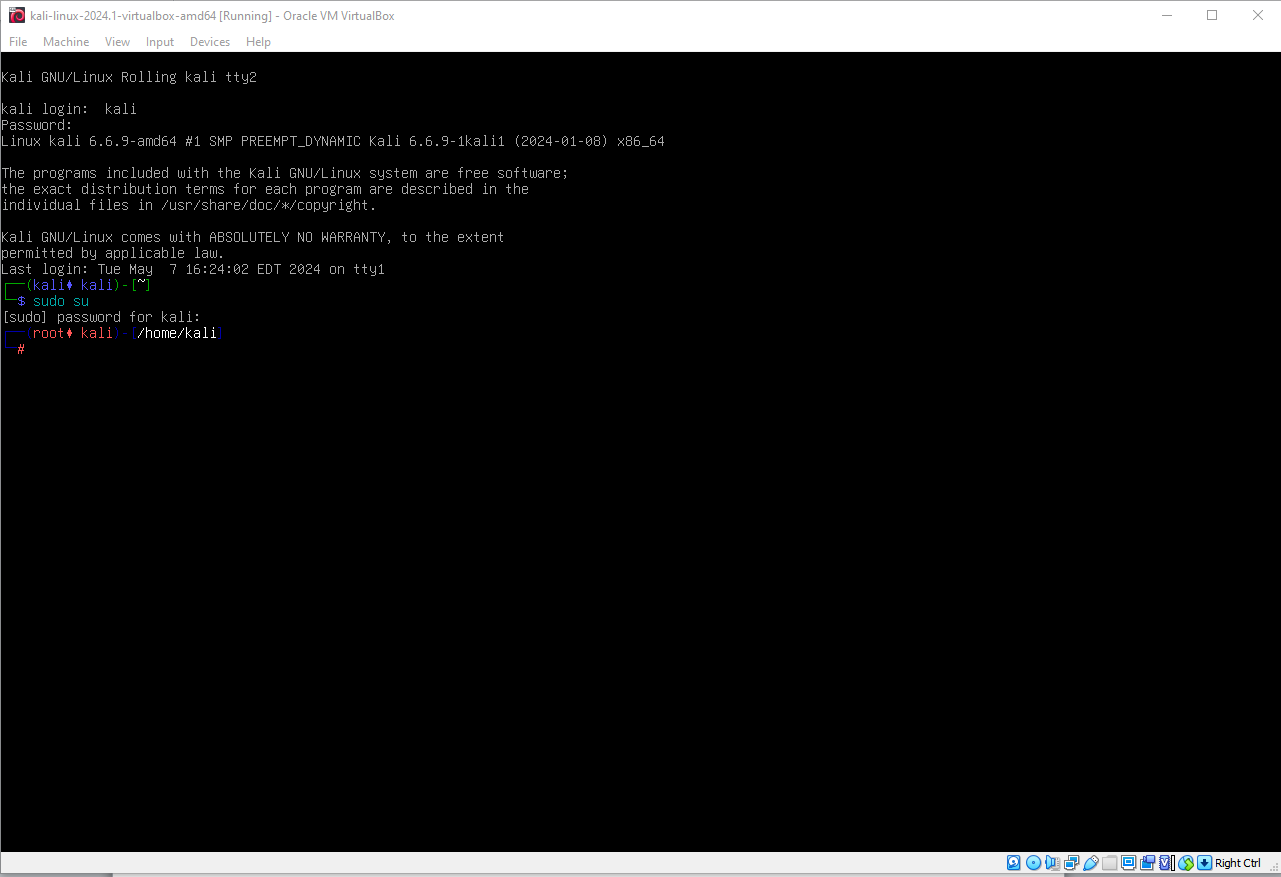
**Задания к лабораторной работе**

Откройте два терминала (в серверных Linux для переключения между терминалами (tty) обычно используется сочетание клавиш Alt+F[1-5]). В одном из них получите права суперпользователя используя команду sudo su

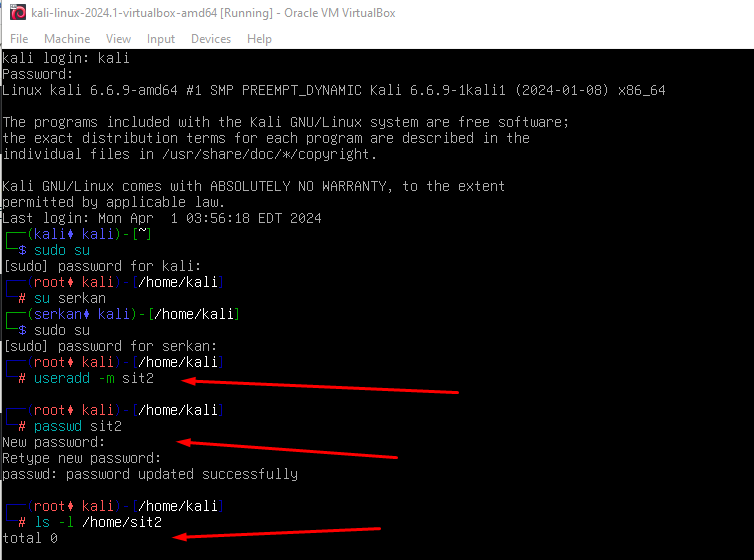
Tty1 - Я ввел команду sudo su



Tty2- Я ввел команду sudo su

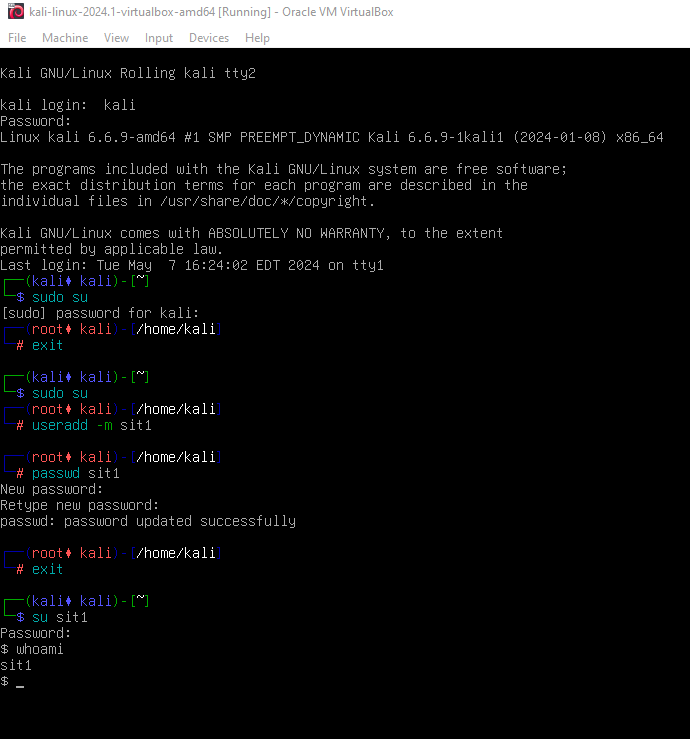
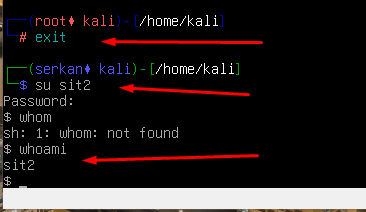


Добавил пользователя «sit2» и пароль



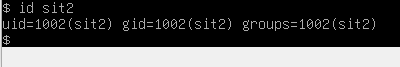
Выйдите из суперпользователя командой exit

Войдите под первым терминалом в пользователя «sit», во втором в пользователя «sit2».

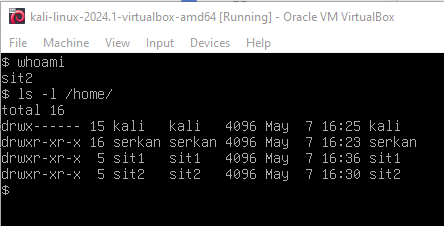


Посмотрите какой идентификатор получил пользователь «sit» и пользователь «sit2» используя команду id

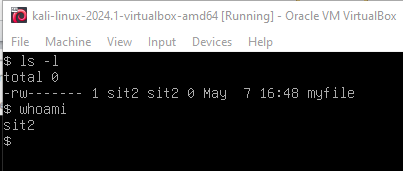




Посмотрите права доступа на домашний каталог пользователей «sit» и «sit2», используя команду ls



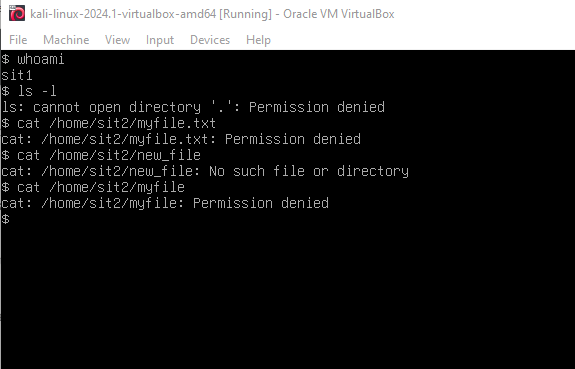
Создайте файл под пользователем «sit2» с маской 0077 используя umask



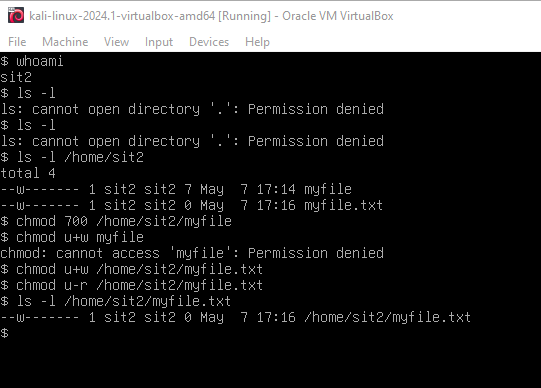
-rw------- - это права доступа к файлу (rw- для владельца и --- для группы и остальных пользователей), что соответствует маске 0077.

1 sit2 sit2 - это владелец (sit2) и группа владельца (sit2) файла.

0 - это размер файла (в данном случае файл пустой).

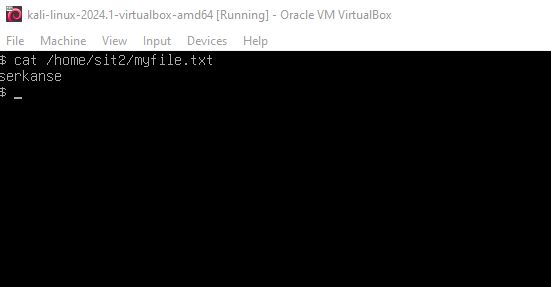
Попробуйте прочитать его содержимое под пользователем «sit» используя команду cat

**sit** нет разрешения на чтение содержимого файла **myfile.txt** из-за ограниченных прав доступа.

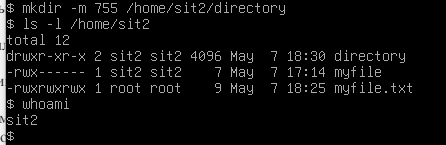
Измените права доступа на файл так, чтобы пользователь «sit» мог записывать в файл, но не читать его.

**u+w** для установки прав на запись для пользователя **sit**:  
**u-r** для снятия прав на чтение у пользователя **sit**:

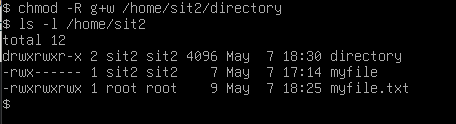
Запишите текстовую информацию в файл из под пользователя «sit» используя консольный текстовый редактор vi или nano



Создайте каталог из под пользователя «sit2»

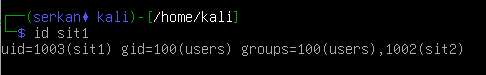


Установите права записи для группы пользователей на данный каталог

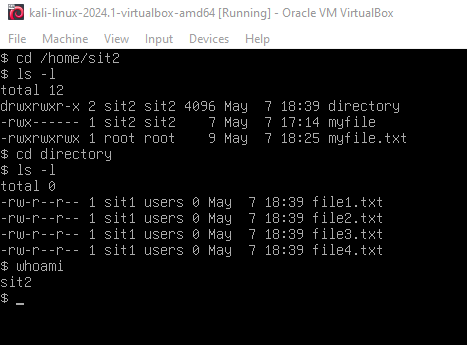


Добавьте пользователя «sit» в группу «sit2» с помощью команды usermod

Проверьте в какие группы входит пользователь «sit»



Создайте несколько файлов в каталоге, который был создан пользователем «sit2» из под пользователя «sit».



Ознакомьтесь как удалить пользователя вместе с содержимым его домашнего каталога из справочной документации

Удалите пользователя «sit2» вместе с его домашним каталогом.

