Разграничение доступа субъектов к объектам.

**Избирательное управление доступом** (англ. *discretionary access control*, *DAC*) — управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа. Также используются названия *дискреционное управление доступом*, *контролируемое управление доступом* и *разграничительное управление доступом*.

Субъект доступа «Пользователь № 1» имеет право доступа только к объекту доступа № 3, поэтому его запрос к объекту доступа № 2 отклоняется. Субъект «Пользователь № 2» имеет право доступа как к объекту доступа № 1, так и к объекту доступа № 2, поэтому его запросы к данным объектам не отклоняются.

Для каждой пары (субъект — объект) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т. д.), то есть тех типов доступа, которые являются санкционированными для данного субъекта (индивида или группы индивидов) к данному ресурсу (объекту).

**Управление доступом на основе ролей** (англ. *Role Based Access Control, RBAC*) — развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.

Формирование ролей призвано определить чёткие и понятные для пользователей компьютерной системы правила разграничения доступа. Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы правила разграничения доступа.

Такое разграничение доступа является составляющей многих современных компьютерных систем. Как правило, данный подход применяется в системах защиты СУБД, а отдельные элементы реализуются в сетевых операционных системах. Ролевой подход часто используется в системах, для пользователей которых чётко определён круг их должностных полномочий и обязанностей.

Несмотря на то, что *Роль* является совокупностью прав доступа на объекты компьютерной системы, ролевое управление доступом отнюдь не является частным случаем избирательного управления доступом, так как его правила определяют порядок предоставления доступа субъектам компьютерной системы в зависимости от имеющихся (или отсутствующих) у него ролей в каждый момент времени, что является характерным для систем мандатного управления доступом. С другой стороны, правила ролевого разграничения доступа являются более гибкими, чем при мандатном подходе к разграничению.

**Мандатное управление доступом** (англ. *Mandatory access control, MAC*) — разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. Также иногда переводится как **Принудительный контроль доступа**. Это способ, сочетающий защиту и ограничение прав, применяемый по отношению к компьютерным процессам, данным и системным устройствам и предназначенный для предотвращения их нежелательного использования.

Согласно требованиям ФСТЭК, мандатное управление доступом или «метки доступа» являются ключевым отличием систем защиты государственной тайны РФ старших классов 1В и 1Б от младших классов защитных систем на классическом разделении прав по матрице доступа.

В настоящее время информационные системы имеют либо встроенные инструменты разграничения доступа к объектам доступа, либо позволяют использовать интегрируемые решения для управления доступом.

Далее на примере Unix OS представлен метод разграничения доступа к файлам и каталогам.

Операционные системы Unix - это многопользовательские системы, которые дает огромные возможности манипулирования доступом к данным для каждого пользователя отдельно. Это позволяет гибко регулировать отношения между пользователями, объединяя их в группы, что позволит защитить данные одного пользователя от нежелательного вмешательства других. В UNIX-подобных системах файлы также обеспечивают доступ к периферийным устройствам, дисковым накопителям, принтерам и т.п.

**Запуск виртуальной машины с образом Unix-системы**

С сайта <https://www.virtualbox.org/wiki/Downloads> скачивается и устанавливается оснастка для виртуальных машин под вашу операционную систему. Установка проходит интерактивно по шагам: 

С сайта, предоставляющего дистрибутивы образы операционных систем, берем интересующий образ OS семейства Unix. Ссылки на официальные источники с дистрибутивами:

<https://www.kali.org/get-kali/#kali-platforms> - Kali

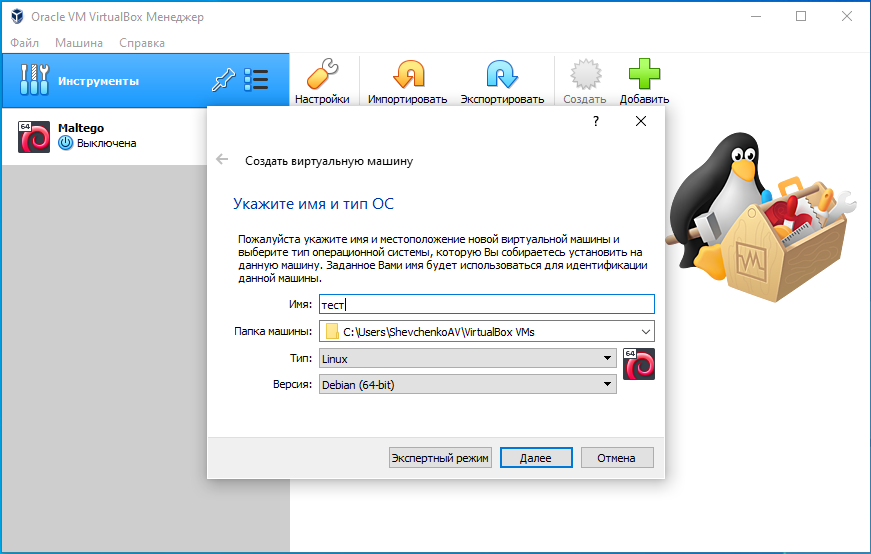
<https://ubuntu.com/download/desktop> - Ubuntu

<https://www.centos.org/download/> - CentOS

<https://www.debian.org/index.ru.html> - Debian

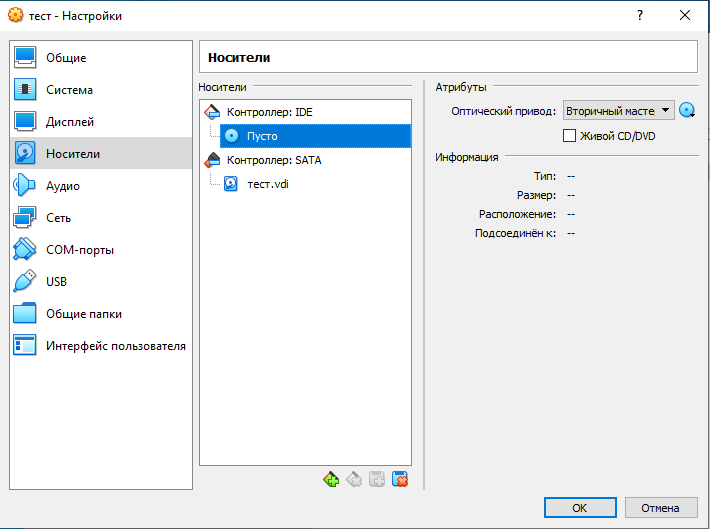
Для выполнения задания подойдет любая из перечисленных операционных систем.

В оснастке VirtualBox в меню Машина выбираем Создать. В появившемся окне по шагам заполняем поля и указываем образ создаваемой виртуальной машины.



Параметры объема оперативной памяти, объема выделяемого жестка диска необходимо выставлять исходя из возможностей вашей рабочей станции. Можно использовать предлагаемые установщиком, но это скажется на скорости работы создаваемой виртуальной машины. Для целей настоящей лабораторной работы высокая производительность не требуется.

После подготовки виртуальной машины необходимо установить скаченный ранее образ в виртуальный привод. В настройках созданный виртуальной машины выберите вкладку Носители и установите для контроллера IDE соответствующий носитель.

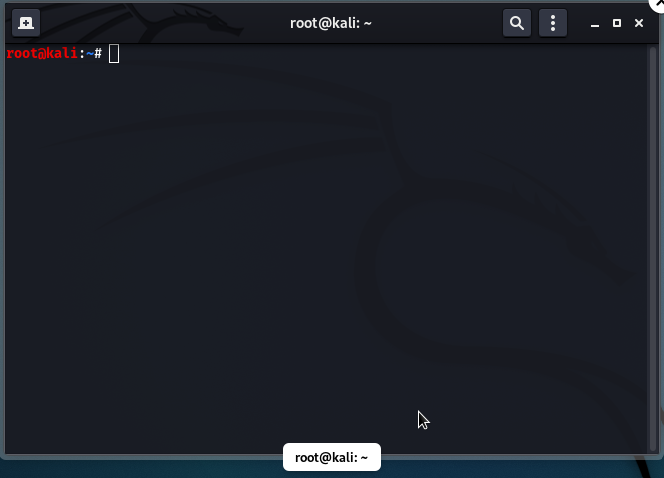


После проделанный операций запустите вашу виртуальную машину и пройдите поэтапно интерактивный процесс установки Unix-системы. Большая часть параметров будет предустановлена по умолчанию. Вам будет необходимо указать логин и пароль пользователя, которые будут использоваться для вашей аутентификации.

**Терминал в Unix-системе**

В обозначенных выше Unix-системах присутствуют режим взаимодействия с пользователем посредством командной строки.

Все нижеприведенные команды вводятся в командной строке. Туда же выводится результат выполнения команд.

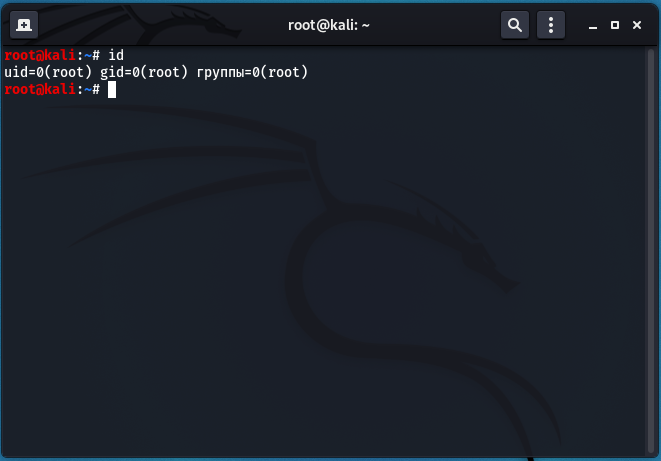


**UID, GID**

Каждый пользователь в системе имеет свой уникальный идентификационный номер (user-ID, или UID). Также пользователи могут объединяться в группы, которые в свою очередь имеют group-ID, или GID. Чтобы узнать свой UID и GID, т.е. уникальный номер пользователя и номер группы, к которой Вы принадлежите, необходимо ввести команду id:

root@kali:~# id

uid=0(root) gid=0(root) группы=0(root)



В данном примере представлен пользователь root (пользователь с наивысшим приоритетом работы и максимальным доступом). В современных системах чаще всего пользователи работают из под своих индивидуальных учетных записей, и, соответственно, их имя учетной записи будет другим, как и соответствующие параметры при выводе команды id.

**Создание учетной записи**

Для создания новой учетной записи чаще всего используется команда adduser.

Команда adduser используется для упрощения создания учетных записей пользователей, а также автоматической проверки их соответствия политике дистрибутива. Она реализована в виде обертки для таких утилит, как useradd, passwd и chfn и использует дополнительный файл конфигурации /etc/adduser.conf для хранения параметров, относящихся к политике дистрибутива.

В зависимости от используемого дистрибутива операционной системы могут быть незначительные отступления в логике создания пользователя, но, чаще всего, команда на создание пользователя будет одна и та же, как и передоваемые ей параметры и значения.

Базовый синтаксис команды adduser:

# adduser <имя пользователя>

В результате исполнения данной команды выполняются следующие операции:

Создается учетная запись пользователя с именем <имя пользователя>.

Создается домашняя директория пользователя /home/<имя пользователя>, в которую копируются файлы из директории /etc/skel.

Создается группа пользователей с именем, соотвествующим имени нового пользователя, причем сам этот пользователь добавляется в нее.

Выводится запрос ввода пароля для учетной записи пользователя.

Выводится запрос ввода дополнительной информации, относящейся к учетной записи пользователя.

Несложно догадаться, что для выполнения практически всех описанных действий может использоваться приведенная ниже последовательность команд на основе более распространенных утилит:

# useradd -m –U <имя пользователя>  
# passwd <имя пользователя>  
# chfn <имя пользователя>

Еще одним вариантом использования рассматриваемой команды является добавление существующего пользователя в новую группу пользователей. В этом случае используется следующий синтаксис:

# adduser <имя пользователя> <имя новой группы пользоваетелей>

Данная команда также может быть заменена на более сложную команду на основе более распространенной утилиты usermod:

# usermod -G <имя старой группы пользоваетелей>, ..., <имя новой группы пользоваетелей>

Вторая команда является более сложной, но при этом и более гибкой, ведь она позволяет добавить пользователя сразу в несколько новых групп.

Примеры использования:

1. Создание учетной записи пользователя test\_user

# adduser test\_user

1. Добавление пользователя test\_user в группу test\_user\_group

# adduser test\_user test\_user\_group

Чтобы начать работать от имени пользователя username в Linux просто наберите в терминале команду:

# su - username

и нажмите клавишу Enter. При смене пользователя система может запросить у вас пароль для подтверждения полномочий. Исключение, когда вы переходите от более привилегированного пользователя к мене привилегированному.

чтобы войти под root без смены среды окружения используйте команду:  
# su

чтобы войти под root со сменой смены среды окружения (добавьте тире):

# su -

**Права доступа к файлам**

В свою очередь файлы имеют двух владельцев: пользователя (user owner) и группу пользователей (group owner). Для каждого файла есть индивидуальные права доступа, которые разбиты на три группы:

1. Доступ для пользователя-владельца файла (owner).

2. Доступ для группы-владельца файла (group).

3. Доступ для остальных пользователей (others).

Для каждой категории устанавливаются три вида доступа: (x) - право на запуск файла, (r) - право на чтение файла, (w) - право на изменение (редактирование) файла. Для того, чтобы увидеть права доступа к файлам необходимо ввести команду ls с ключом -l:

root@kali:~# ls -l /home/file.tmp -rwxr-xr-- 1 dmitry users 33 Dec 1 00:38 file.tmp

Для данного примера мы видим, что владелец имеет права на чтение, запись, и выполнение (первые три буквы rwx), группа пользователей может лишь читать и выполнять этот файл (следующие три r-x), ну а остальные пользователи могут только читать данный файл (последние символы r--).

**Изменение прав доступа**

Права пользователя могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда

chmod {параметр доступа} name1 [name2...]

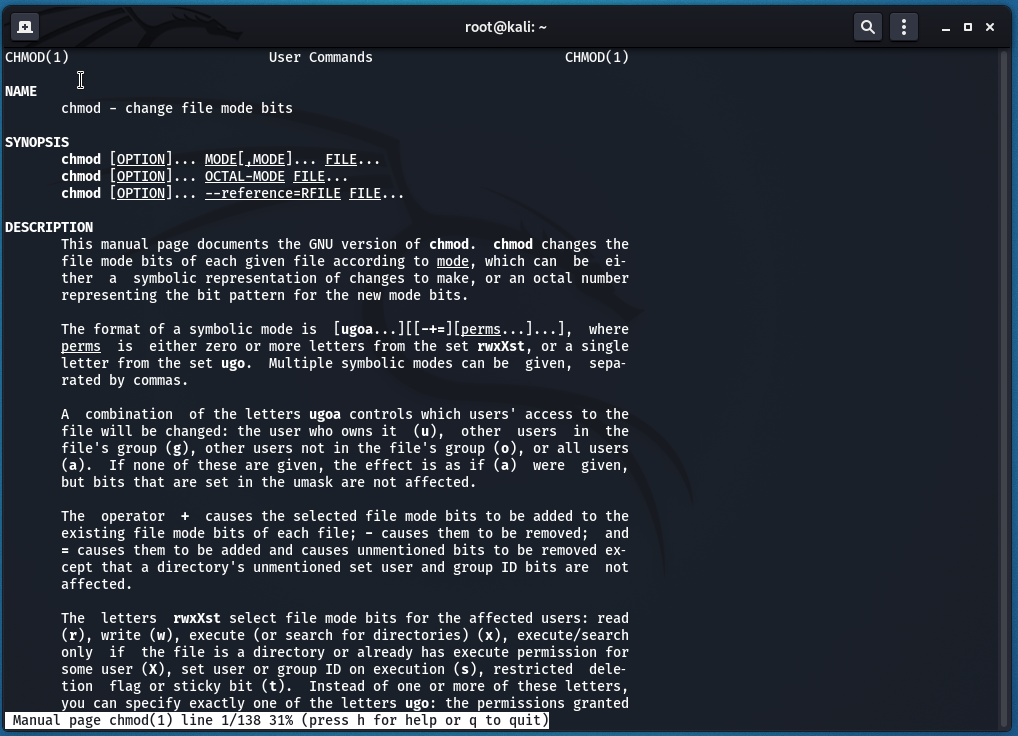
В качестве аргументов команда принимает указание классов доступа ('u' - владелец-пользователь, 'g' - владелец-группа, 'o' - остальные пользователи, 'a' - все вышеперечисленные группы вместе), права доступа ('r' - чтение, 'w' - запись, 'x' - выполнение) и операцию, которую необходимо произвести ('+' - добавить, '-' -убрать, '=' - присвоить). Таким образом, чтобы разрешить выполнение файла prog.pl всем пользователем необходимо выполнить команду:

root@kali:~# chmod a+x prog.pl

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить:

root@kali:~# chmod go-w prog.pl

Полный доступ к описанию возможностей команды chmod доступен в руководстве, вызываемом командой man chmod:



Рассмотрим еще несколько примеров:

# chmod go=w prog.pl

Данная команда позволяет установить право на запись для всех пользователей кроме владельца

# chmod a+x prog.pl

Данная команда позволяет предоставить право на запись для всех пользователей

# chmod g+x-w prog.pl

Данная команда позволяет добавить для группы право на выполнения файла, но снять право на запись.

Права доступа можно представить в виде битовой строки, в которой каждые 3 бита определяют права доступа для соответствующей категории пользователей, как представлено в таблице:

|  |  |  |
| --- | --- | --- |
| rwx | rwx | rwx |
| 421 | 421 | 421 |
| user | group | others |
| владелец | группа | остальные |

Полный список числовых обозначений для применения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OCT** | **BIN** | **Mask** | **Права на файл** | **Права на каталог** |
| 0 | 000 | - - - | отсутствие прав | отсутствие прав |
| 1 | 001 | - - x | права на выполнение | доступ к файлам и их атрибутам[1)](https://help.ubuntu.ru/wiki/%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%B0_unix#fn__1) |
| 2 | 010 | - w - | права на запись | отсутствие прав |
| 3 | 011 | - w x | права на запись и выполнение | все, кроме доступа к именам файлов[2)](https://help.ubuntu.ru/wiki/%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%B0_unix#fn__2) |
| 4 | 100 | r - - | права на чтение | только чтение имен файлов |
| 5 | 101 | r - x | права на чтение и выполнение | чтение имен файлов и доступ файлам и их атрибутам[3)](https://help.ubuntu.ru/wiki/%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%B0_unix#fn__3) |
| 6 | 110 | r w - | права на чтение и запись | только чтение имен файлов |
| 7 | 111 | r w x | полные права | все права |

Для администрирования часто удобнее использовать не буквенное представление прав, а цифровое, в восьмеричном представлении (оно короче). Так, например, права на файл всем и вся, соответствуют записи 777 (что аналогично символьному представлению **rwxrwxrwx**).

Таким образом, для команды chmod 666 prog.pl имеем:

root@kali:~# chmod 666 prog.pl

root@kali:~# ls -l prog.pl

-rw-rw-rw- 1 dmitry users 78 Nov 20 prog.pl

Команда **chmod 644** *имя\_файла* устанавливает "обычные" права доступа, т.е. владелец может читать и записывать в файл, а все остальные пользователи - только читать.

**Особенности прав доступа для каталогов**

Права доступа для каталогов не столь очевидны. Это в первую очередь связано с тем, что система трактует операции чтения и записи для каталогов отлично от остальных файлов. Право чтения каталога позволяет Вам получить имена (и только имена) файлов, находящихся в данном каталоге. Чтобы получить дополнительную информацию о файлах каталога (например, подробный листинг команды ls -l), системы придется "заглянуть" в метаданные файлов, что требует права на выполнения для каталога. Право на выполнение также потребуется для каталога, в который Вы захотите перейти (т.е. сделать его текущим) с помощью команды cd.

**T-бит, SUID и SGID**

Наиболее внимательные пользователи быстро замечают, что помимо стандартных "rwx" значений существуют еще и буквы "s" и "t".

В действительности, битовая маска прав доступа к файлам содержит 4 группы по 3 бита в каждой. Таким образом, команда chmod 755 - это всего лишь краткая запись полной формы команды: chmod 0755.

t-бит обычно используется с каталогами. Обычно, когда t-бит для каталога не установлен, файл в данном каталоге может удалить любой пользователь, имеющий доступ на запись к данному файлу. Устанавливая t-бит на каталог мы меняем это правило таким образом, что удалить файл из каталога может только владелец этого каталога или файла. Установить t-бит можно при помощи команд chmod a+tw somefile или chmod 1777 somefile.

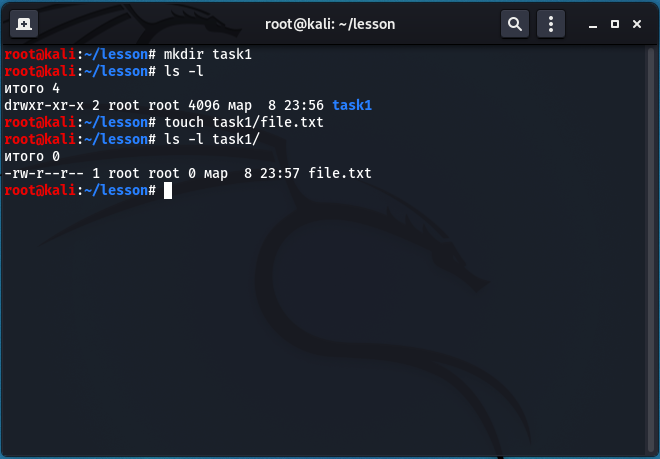
Атрибуты SUID и SGID позволяют изменить права пользователя при запуске на выполнения файла, имеющего эти атрибуты. Запускаемая программа получает права доступа к системным ресурсам на основе прав доступа пользователя, запустившего программу. Установка же флагов SUID и SGID изменяет это правило таким образом, что назначает права доступа к системным ресурсам исходя из прав доступа владельца файла. Т.е. запущенный исполняемый файл, которым владеет суперпользователь, получает права доступа к системным ресурсам на уровне суперпользователя (фактически неограниченные). При этом установка SUID приведет к наследованию прав владельца-пользователя файла, а установка SGID -владельца-группы. В завершении хочется отметить, что пользоваться такими мощными атрибутами как SUID и SGID нужно с крайней осторожностью, особенно подвергать пристальному вниманию программы и скрипты, владельцем которых является root (суперпользователь), т.к. это потенциальная угроза безопасности системы.

Задание1:

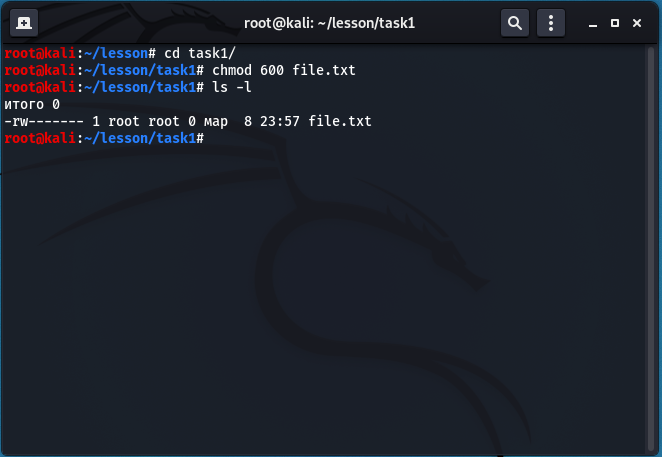
1. В виртуализированной среде Unix (например, Ubuntu или Debian) в домашнем каталоге создайте каталог (в примере task1) и поместите в него пустой текстовый файл (file.txt) с помощью команд **mkdir** и **touch**. Файл также можно создать перенаправив поток вывода команды **echo** и указателя **>** или **>>**:

# echo > file.txt

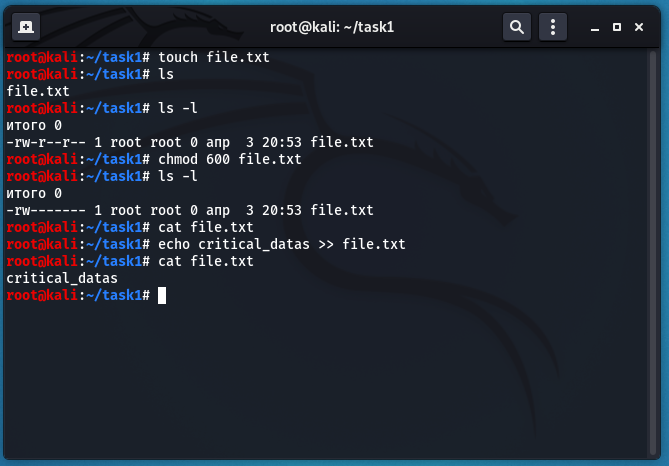
Выполните в созданном каталоге команду ls –l, чтобы отобразить права на созданный файл.



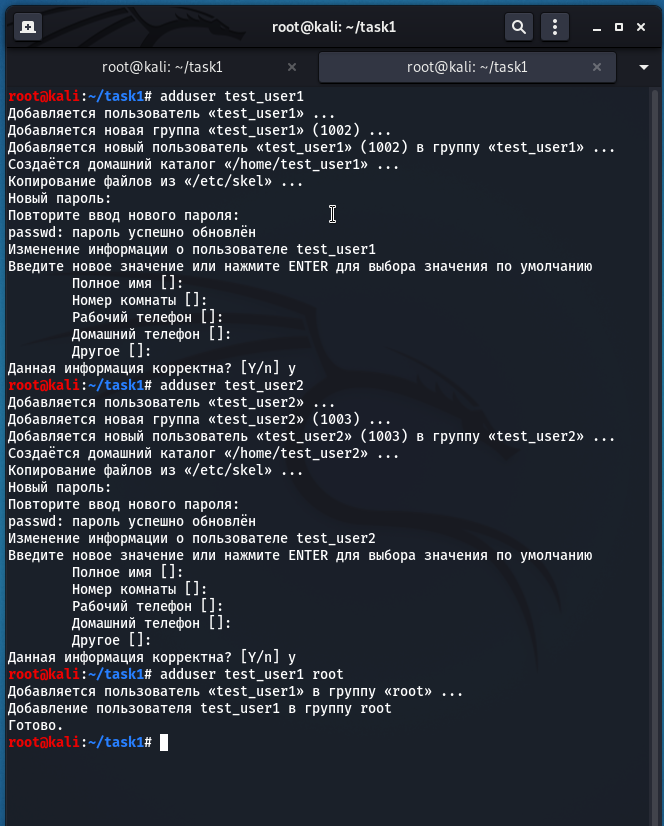
Смените права на файл командой **chmod**, чтобы был доступ на “чтение и запись” только у данного пользователя, и не было доступа у группы пользователей и прочих пользователей. Результат отобразите командой ls –l.



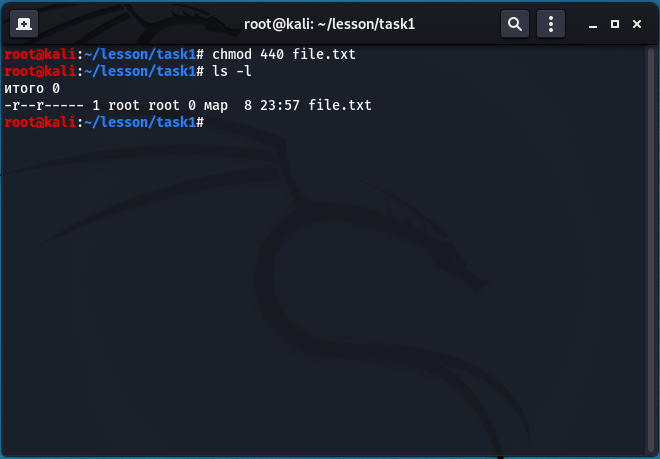
Попробуйте записать любые данные от имени пользователя в текстовый документ, после изменения прав доступа, используя команду **echo** и перенаправив вывод в ваш файл путем дозаписи **>>**.



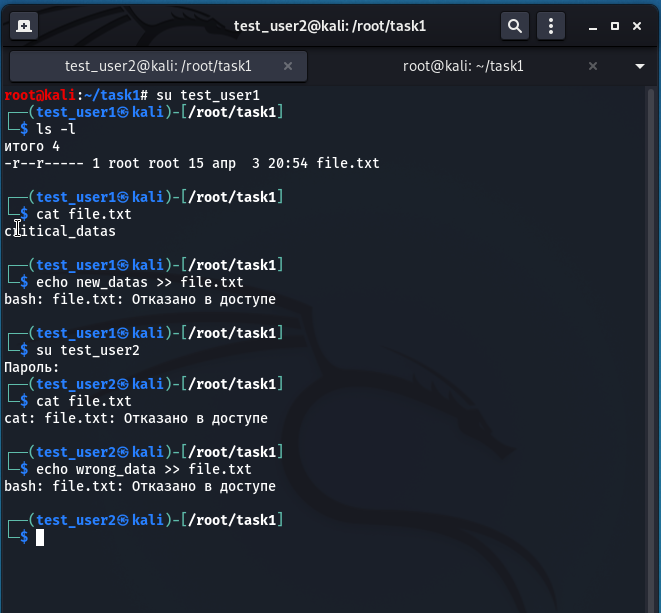
Создайте второго пользователя в системе и добавьте его в ту же группу для совместной работы, создайте третьего пользователя, но не изменяйте его базовые настройки. В приведенном ниже примере был добавлен пользователь test\_user1 и test\_user2. После чего пользователь test\_user1 был добавлен в группу root.



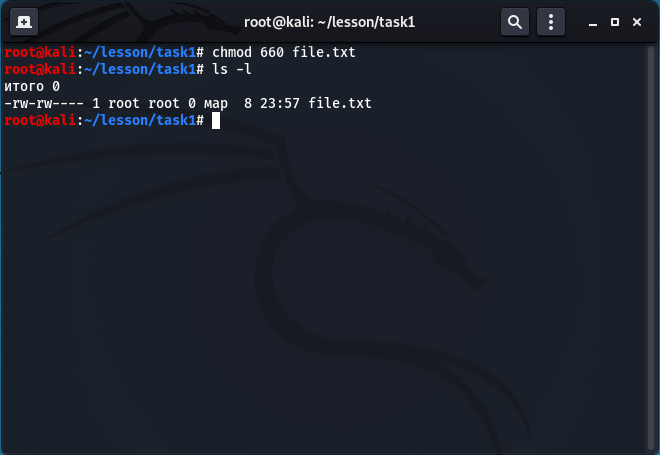
Измените права на доступ к файлу для пользователя, группы пользователей “только на чтение”.



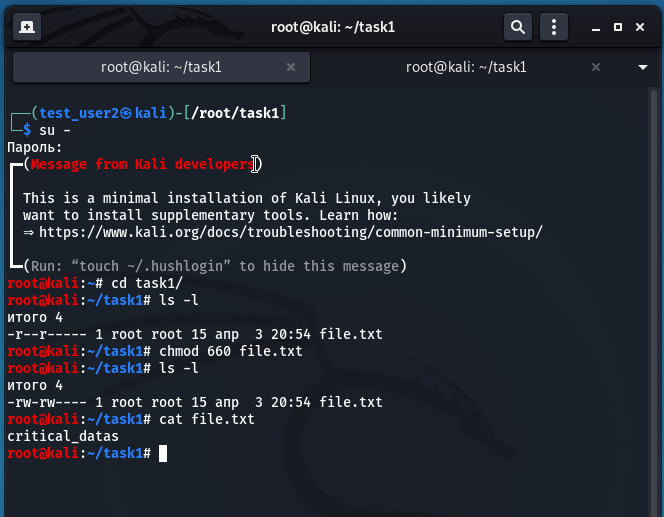
Смените активного пользователя на вновь созданного пользователя с той же группой, попробуйте обратиться к файлу на чтение и запись. Проделайте эти же обращения от имени последнего созданного пользователя. Обратите внимание на вывод по результату обращений.

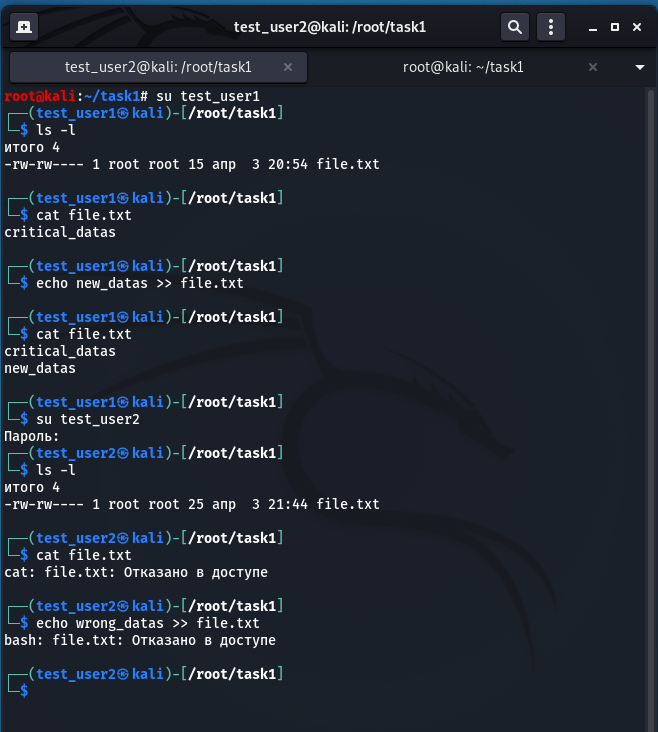


Измените права на доступ к файлу для пользователя, группы пользователей “чтение и запись”.

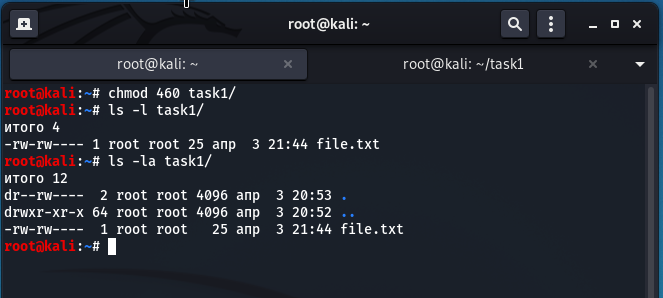


Смените активного пользователя на вновь созданного пользователя с той же группой, попробуйте обратиться к файлу на чтение и запись. Проделайте эти же обращения от имени последнего созданного пользователя. Обратите внимание на вывод по результату обращений.

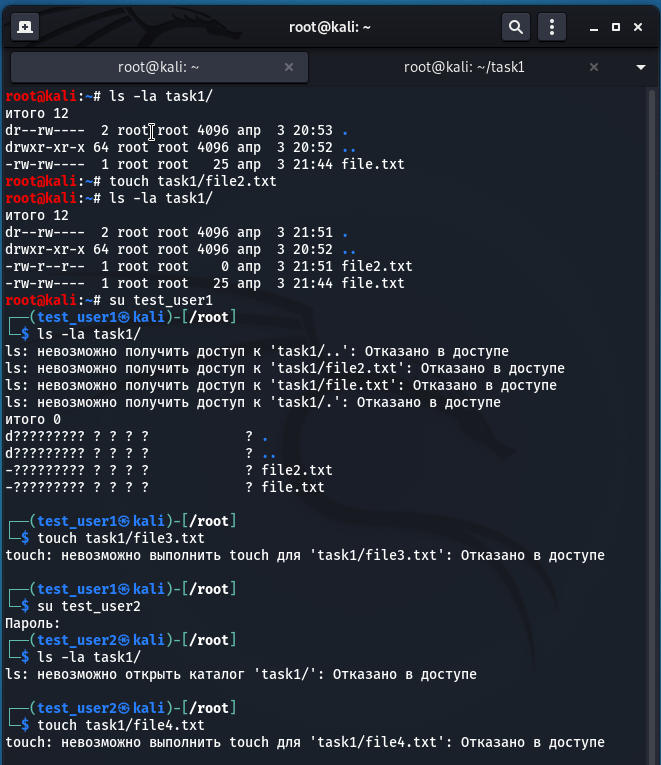




От имени администратора системы смените права на каталог на “только чтение” для пользователя, “чтение и запись” для группы, а для других пользователей “никаких прав”. Отобразите содержимое каталога командой ls -la. Обратите внимание, использован дополнительный флаг “a”.



Попробуйте создать новый файл и отобразить содержание каталога с нашим файлом от имени каждого из трех пользователей. Обратите внимание на вывод результата команд.



Проделайте описанные шаги, сформируйте отчет. Приложите к отчету скриншоты экрана и команды для командной строки.