Управление правами пользователей

Использование учетной записи непосредственно связано с наличием или отсутствием тех или иных прав доступа к фалам системы. Изменим прав доступа может заниматься только суперпользователь. Для настройки прав доступа к файлам или каталогам используется команды

- *chgrp* – изменение принадлежности файла или каталога к определенной группе;
- *chown* – изменяет владельца файла или каталога;
- *chmod* – изменяет режим доступа к файлу или каталогу.
- *umask* – задает права доступа к вновь создаваемым файлам.

Права доступа бывают трех видов:

- чтение (*read, r*), *ID* 4;
- запись (*write, w*), *ID 2;*
- выполнение (*execute, x*) , *ID1*.

Как правило используется команда *chmod,* она имеет следующий синтаксис.

chmod [u|g|o|a] [+|-|=] [r|w|x] dir or file 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, например запрос

chmod a+x testdir/testfile

позволяет выполнять файл testfile всем пользователям.

А, например, запрос вида

chmod go-w testdir/testfile

Оставляет права записи только за владельцем файла.

В действительности команда *chmod* имеет битовые значения прав доступа (*4,2,1*) для каждого из трех видов пользователей (*u,g,o*) поэтому права доступа к команде могут быть заданы альтернативно в виде битовой маски вида, например, *111* – сделать файл выполняемым для всех трех групп или, например, *777* (*4+2+1* для каждой группы) – то есть разрешить любые действия с файлом всем.

Использование списков контроля доступа.

Описанные выше команды позволяют полностью настроить особенности доступа к конкретным файлам или папкам. Однако при их использовании могут возникнуть сложности с настройкой таковых прав для отдельных пользователей. Для решения таких пользователей в современных версиях *Linux* поддерживается политика списков доступа (*access control list, ACL*). Списки доступа могут включать специализированные права доступа для заданных файлов или каталогов, или права доступа по умолчанию. Список доступа по умолчанию является необязательным. При этом права доступа устанавливаются при помощи команды

#setfacl -m <type>:usrer_name:permissions file_name

где *type – user (u),group (g) ,other (o),* а также mask (для всех, кроме владельца и other); *permissions – rwx.*

Также допустимы записи вида *#setfacl -m <type>::permissions file_name*

Для установки прав доступа по умолчанию (только для каталогов) вместо *<type>* указывается *d (default).*

Узнать текущие права доступа к файлам или каталогом можно при помощи команды *getfacl.*

Задание 1:

1. Создайте директорию *test* (команда *mkdir*)
2. Создайте файл *tetst* (команда *touch*)
3. Узнать текущие атрибуты/права доступа к файлу *tetst* при помощи команды *ls –la*.

   Также права доступа к фалам (но не директориям) могут быть получены при помощи команды *ll*.
4. Измерите права доступа к файлу на только чтение для всех
5. Измените права доступа к файлу на только исполнение и только для владельца группы.
6. Добавьте права доступа к файлу на исполнение для владельца группы.
7. Измените владельца файла.
8. Измените принадлежность файла к группе.
9. Установить право на запись файла для всех пользователей кроме владельца.
10. Задайте права доступа: любые права, чтение и редактирование и никаких прав для видов пользователей соответственно (*u,g,o),* в виде одной команды в форме битовой макси.
11. Ссоздайте жесткую и символическую ссылку на исходный файл:

    $ ln file ./hardlink и *$ ln -s file ./symlink*

Символические ссылки похожи на ярлыки в Windows. Они содержат адрес нужного файла в вашей файловой системе. Когда вы пытаетесь открыть такую ссылку, то открывается целевой файл или папка.

Жёсткие ссылки ссылаются на участок жесткого диска, файл на который имеется ссылка можно перемещать между каталогами, и все ссылки останутся рабочими, поскольку для них неважно имя.

Выполните команду: *$ ls -la* объясните различие в правах доступа к исходному файлу и ссылкам на этот файл. Измените права доступа к жесткой ссылке (*hardlink)*. Как эта операция отразилась на самой жесткой ссылке и на исходном файле? Повторите те же действия для символической ссылки (*symlink*). Проанализируйте результат. В чем отличия между первым и вторым случаем?

12. Повторите пункты 3-6,9,10 при помощи функционала *ACL*.

Задание 2:

1. Создайте несколько пользователей.
2. Выполните следующие команды от лица первого пользователя:

$ mkdir x
$ chmod 771 x
$ mkdir w
$ chmod 772 w
$ mkdir wx
$ chmod 773 wx
$ mkdir r
$ chmod 774 r
$ mkdir rx
$ chmod 775 rx
$ mkdir rw
$ chmod 776 rw
$ mkdir rwx
$ chmod 777 rwx

3. Убедитесь в том, что каталоги созданы и имеют соответствующие права доступа. Какие из предоставленных прав кажутся вам лишенными смысла? Почему?
4. От лица второго пользователя для каждого каталога создайте файл командой

$ echo `date` > /study/lab3/part3/<dir_name>/file

Повторите попытку для пользователя первого пользователя и суперпользователя. Результат выполнения команды зафиксируйте в таблице вида

	x	*w*	*wx*	*r*	*rx*	*rw*	*rwx*
root							
User1							
User2							

Для каких каталогов выполнение команды завершилось неудачно? Почему?

5. Для каждого каталога выполните следующие команды:
   $ ls -la /study/lab3/part3/<dir_name>/
   $ cat /study/lab3/part3/<dir_name>/file
6. В чем заключается отличие в результате выполнения команд для разных каталогов? Чем оно объясняется?

Задание 3.

1. Узнайте текущую маску пользователя (по умолчанию) при помощи команды *umask.*
2. Задайте *umask* 777, создайте файл и отдельно папку.
3. Попробуйте зайти в папку от лица пользователя.
4. Попробуйте зайти в паку от лица суперпользователя.
5. Какие права доступа у созданных файла и папки?.
6. Задайте *umask 000* повторите предыдущие пункты, отразите в отчете как работает команда umask.

Задание 4:

Используя полученные знания об учетных записях пользователей, группах и правах доступа к объектам файловой системы предложите решение для следующей практической задачи.

Политика безопасности организации предполагает использование информации следующих уровней конфиденциальности:

- Открытая информация (уровень 3)
- Внутренняя информация (уровень 2)
- Конфиденциальная информация (уровень 1)

Каждый сотрудник подразделения работает в контексте своего домашнего каталога (*/home/<username>*).

Сотрудники с 3-м уровнем доступа работают только с открытой информацией. Они имеют право просматривать (но не изменять!) файлы только своих коллег с аналогичным уровнем доступа.

Сотрудники со 2-м уровнем доступа имеют доступ на чтение к домашним каталогам 2-го и 3-го уровней.

Сотрудники с высшим уровнем доступа имеют доступ на чтение к домашним каталогам 2-го и 3-го уровней, но не к каталогам друг друга.

Руководитель подразделения имеет доступ на чтение к файлам всех своих подчиненных.

Соответствие учетных записей сотрудников их уровню доступа приведены в таблице.

пользователь	Уровень доступа			
	руководитель	1	2	3
0-1	x			
1-1		x		
1-2		x		
2-1			x	
2-2			x	
3-1				x
3-2				x

Зафиксируйте полученный результат и сделайте вывод.