Управление правами доступа

Цель работы

Получить навыки по управлению правами доступа к файловым ресурсам системы, освоить основные команды - *chmod, chown, chgrp*.

Задания к работе

1. Войти в систему с собственной учетной записью
2. Создать в домашнем каталоге 2-3 файла произвольного содержания (имена файлов - u1, u2, u3).
3. Получить развернутый список файлов домашнего каталога и сохранить его в файле listing1
4. Просмотреть файл listing1, обратив внимание на поля прав доступа, владельца и группы
5. Повторить п. 2 от имени пользователя root в новом сеансе или по команде su (имена файлов - r1, r2, r3). Завершить сеанс root
6. Повторить п.3, результат дописать в файл listing1
7. Открыть файл listing1 и сравнить права доступа для файлов, созданных от вашего имени и от имени суперпользователя
8. Изменить содержимое файлов, созданных вами и суперпользователем. Сохранить изменения
9. Открыть сеанс root
10. Перейти в каталог /home/ваша_учетная_запись
11. Изменить права доступа к файлам u1 и r1 следующим образом:
    - u1: запретить запись для владельца и группы
    - r1: разрешить запись для всех
12. Переключиться в сеанс пользователя и изменить содержимое файлов u1 и r1. Сохранить изменения
13. Перейти в сеанс root и изменить владельца файлов u1 и u2 на root, а группу - на stud

14. Из tty1 попробовать изменить файл u2
15. Из tty2 создать каталоги /home/shared, home/shared/pub, /home/shared/upload, /home/shared/temp. Установить на них следующие права:

каталог	владелец	группа	права
pub	root	users	775
upload	nobody	users	130
temp	stud	users	777

16. Выполнить копирование, чтение, удаление файлов u1, u2, u3, r1, r2, r3 в каталоги, созданные в п. 17 из сеансов root, stud и вашего. Сравнить и проанализировать результаты.

Методические указания

Выполняя предыдущие лабораторные работы вы уже сталкивались с разграничением прав доступа в ОС Линукс. Такое разграничение обусловлено многозадачностью и многопользовательским режимом Линукс и призвано повысить безопасность и надежность системы, а также обеспечить защиту конфиденциальной информации.

Каждый файл в Линукс характеризуется набором атрибутов, определяющих его принадлежность и права доступа. Отношение принадлежности файла определено для:

- владельца файла(user) - пользователя, создавшего (что не обязательно) этот файл;
- группы (group) - в состав которой входит владелец;
- прочих (other) пользователей.

Для каждого объекта в файловой системе Linux существует набор прав доступа, определяющий взаимодействие пользователя с этим объектом. Такими объектами могут быть файлы, каталоги, а также специальные файлы (например, устройства) — то есть по сути любой объект файловой системы. Так как у каждого объекта в Linux имеется владелец, то права доступа применяются относительно владельца файла. Они состоят из набора 3 групп по три атрибута:

- чтение(r), запись(w), выполнение(x) для владельца;
- чтение, запись, выполнение для группы владельца;
- чтение, запись, выполнение для всех остальных.

Атрибуты файла могут быть представлены в символьном или числовом виде. Символьное представление атрибутов - это строка, где последовательно записаны права доступа в следующем виде:

```
rwxrwxrwx
```

где каждая тройка символов определяет права на чтение (r), запись (w) и исполнение (x) для соответствующих пользователей (первая тройка - для владельца (user), вторая - для группы (group), третья - для прочих (other).

К правам доступа относятся: чтение (read), изменение (write), исполнение (execute). Понимание этих прав будет различным и зависеть от содержания файла. Наибольшие различия - между обычными (regular) файлами и каталогами. Эти различия приведены в табл. 1.

	Файлы	Каталоги
Чтение	Просмотр содержимого файла (например, текста) в соответствующей программе и возможность его копирования	Обзор списка файлов и возможность копирования каталога (в общем случае, вместе со всем содержимым)
Изменение	Редактирование содержимого файла и его копирование, но не удаление или переименование/перемещение	Обеспечивает возможность записи и удаления файлов
Исполнение	Разрешает запуск программ и сценариев оболочки	Разрешает переход в каталог и перемещение по нему

Таким образом появляется возможность создания так называемых "скрытых" каталогов, когда невозможно получить список файлов, но пользователь точно знающий имя файла может скопировать его из "скрытого" каталога.

Вот пример отображения списка файлов с правами доступа, представленными в символьном виде:

```
aag@stilo:~>                                dir                                -L1
итого                                                                       2722316
-rw-r--r--   1   aag   users   498444757   Ноя   27   16:15   aag.asoiu.tar.gz
drwxr-xr-x     2     aag     users     4096     Июн     1     2007     bin
-rw-r--r--    1    aag    users    26    Фев    20    10:20    description.txt
drwxr-xr-x     5     aag     users     4096     Мар     2     20:01     Desktop
drwx------     2     aag     users     4096     Фев     23     09:50     Documents
drwxr-xr-x     4     aag     users     4096     Фев     28     00:03     downloads
-rwxrwxr-x     1     aag     users     7523     Окт     20     2006     Dz19.jpg
-rw-r--r--    1    aag    users    8336    Фев    24    01:12    httpd.myconf
-rw-r--r--     1     aag     users     20     Фев     25     16:32     index.html
-rw-r--r--    1    aag    users    30296    Фев    23    10:05    logofish.xcf
drwxr-xr-x     2     aag     users     4096     Сен     28     22:53     Music
drwxr-xr-x     3     aag     users     4096     Дек     3     13:45     Projects
drwxr-xr-x     4     aag     users     4096     Фев     26     00:05     public_html
-rw-r--r--     1     aag     users     1088     Фев     20     10:18     readme.txt
drwxr-xr-x     4     aag     users     4096     Фев     27     23:41     scrapbook
-rw------- 1 root root 0 Июн 2 2007 session_mm_cli0.sem
```

Обратите внимание на первые символы в записи прав доступа. В приведенном листинге первый символ d указывает, что файл является каталогом. Признаком специального символьного и блочного устройств являются символы c и b, а для каналов (pipes) соответсвенно p.

Числовое представление прав доступа - это трехзначное число, каждая цифра которого определяет (слева направо) права для владельца, группы и прочих. Права определяются как сумма цифр 4 (чтение), 2 (запись) и 1 (исполнение). Таким образом, например файл f1, создателем которого является user1 и разрешенный для чтения и изменения членам группы users и только для чтения всем прочим, будет иметь следующие атрибуты:

- в символьном виде: rw-rw-r--

- в числовом виде: 664

Вновь создаваемый файл обычно получает права rw-r--r-- (зависит от установок системы и значения umask (см. man umask). Для изменения атрибутов используется команда chmod, которая может принимать как символьное, так и числовое представление атрибутов в качестве параметра. Ниже приведены примеры использования команды:

```
aag@stilo:~>                                dir                                hello.txt
-rw-r--r--    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod go+w hello.txt // разрешить запись для группы и прочих
aag@stilo:~>                                dir                                hello.txt
-rw-rw-rw-    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod ug+x hello.txt // разрешить выполнение для владельца и
группы
aag@stilo:~>                                dir                                hello.txt
-rwxrwxrw-    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod a-x hello.txt // запретить выполнение для всех (a == ugo)
aag@stilo:~>                                dir                                hello.txt
-rw-rw-rw-    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod go-w hello.txt // запретить запись для группы и прочих
aag@stilo:~>                                dir                                hello.txt
-rw-r--r--    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod 755 hello.txt // разрешить чтение и выполнение всем и
запись                                                                  владельцу
aag@stilo:~>                                dir                                hello.txt
-rwxr-xr-x    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~>   chmod   644   hello.txt   //   запретить   выполнение   всем
aag@stilo:~>                                dir                                hello.txt
-rw-r--r--    1    aag    users    17    Map    2    22:32    hello.txt
aag@stilo:~> chmod 711 hello.txt // разрешить только выполнение для группы и
прочих
aag@stilo:~>                                dir                                hello.txt
-rwx--x--x 1 aag users 17 Map 2 22:32 hello.txt
```

Для распределения прав доступа в Linux существует множество команд. Основные из них – это chmod, chown и chgrp. Для смены владельца файла и группы (опционально) используется команда chown, а для смены группы - команда chgrp (см. man chown, man chgrp).

В символьном виде использование команды chmod будет выглядеть следующим образом:

```
                |r|
        |u|     |w|
        |g|  |+| |x|
chmod   |o|  |-| |X| filename,
        |a|  |=| |u|
                |g|
                |o|
```

где: u,g,o,a – установка прав для пользователя, группы, остальных пользователей, всех групп прав доступа соответственно.

+,-,= – добавить, удалить, установить разрешение соответственно.

r,w,x,X,u,g,o – право чтения, записи, выполнения, выполнения если есть такое право еще у какой либо из групп доступа, такие же как у владельца, такие же как у группы, такие же как у остальных пользователей.

filename - Имя файла, у которого изменяются права.

Просмотр разрешений, установленных на файл осуществляется командой ls с ключом -l:

```
[student@ns student]$ ls -l lesson5.txt
 -rw------- 1 student student 39 Nov 19 15:17 lesson5.txt
[student@ns student]$ chmod g+rw lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw---- 1 student student 39 Nov 19 15:18 lesson5.txt
[student@ns student]$ chmod o=u lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-rw-  1  student  student  39  Nov  19  15:18  lesson5.txt  [student@ns
student]$ chmod o-w lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-r-- 1 student student 39 Nov 19 15:19 lesson5.txt
[student@ns student]$_
```

Для использования абсолютного режима необходимо представить права доступа к файлу в виде 3-х двоичных групп. Так например:

rwx r-x r-- будет выглядеть как: 111 101 100

Теперь каждую двоичную группу перевести в 8-ричное число: 111 – 7, 101 – 5, 100 – 4 .

Чтобы задать файлу такие права необходимо выполнить команду:

```
[student@ns student]$ ls -l lesson5.txt
-rw-rw-r-- 1 student student 39 Nov 19 15:19 lesson5.txt
[student@ns student]$chmod 754 lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rwxr-xr-- 1 student student 39 Nov 19 15:19 lesson5.txt
[student@ns student]$_
```

Также предложить им проделать то же самое в символьном виде.

Команда chown (CHange OWNer – сменить владельца) – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь права владельца текущего файла или права root . Синтаксис команды прост:

```
chown username:groupname filename
```

где username – имя пользователя – нового владельца файла; groupname – имя группы – нового владельца файла; filename – имя файла, у которого сменяется владелец.

Имя группы в синтаксисе команды можно не указывать, тогда будет изменен только владелец файла.

Команда chgrp используется для изменения владельца-группы файла. Синтаксис ее таков:

```
chgrp groupname filename
```

где: groupname – имя группы, которой будет принадлежать файл filename – имя изменяемого файла

Имейте в виду, что использовать команды chown и chmod может только пользователь- владелец файла и root , а команду chgrp - пользователь-владелец файла, группа- владелец файла и root .

Существуют еще несколько особых прав, которые могут устанавливаться на файлы и каталоги. О некоторых из них мы поговорим при изучении темы "процессы". Но один рассмотрим сейчас. Это так называемый sticky bit (бит прикрепления).

В первых версиях Юникс этот бит использовался для того, чтобы заставить систему при работе программы оставлять образ ее кода в памяти. Тогда при следующем обращении к программе на ее запуск тратилось намного меньше времени так как чтение кода с устройства более не требовалось. Для файлов и сегодня в Linux осталось прежнее значение этого бита. А вот для каталогов этот атрибут приобрел новое значение. Если sticky bit установлен на каталог, то удалить файлы из такого каталога может только пользователь-владелец файла, и то только если у него есть право на запись в файл. Группа-владелец и остальные пользователи даже при наличии прав на запись в файл не смогут удалить его при установленном на каталог sticky bit. Бит прикрепления устанавливается командой chmod в символьном виде:

```
chmod +t filename
```

Контрольные вопросы

1. Зачем у файла нужны атрибуты доступа?
2. Какие три категории пользователей знают права доступа каждого конкретного файла?
3. Для чего в UNIX-подобных системах используются пользовательские группы?
4. Какие действия с файлами регламентируются правами доступа?
5. На какой системе счисления основывается числовое представление прав доступа?
6. Чем отличается назначение прав доступа к простому файлу и к директории?
7. Кто может менять права доступа определенного файла?
8. Каково назначение специальных битов прав доступа?

Дополнительные задания

1. В текущей папке создать файл hello следующего содержания

```
#!/bin/sh
echo                              Hello,                                World!
echo                        -n                        "I'm                        "
whoami
```

2. Выполнить следующие действия и проанализировать результаты:
    - набрать в командной строке имя файла hello и нажать Enter
    - набрать в командной строке sh hello и нажать Enter
    - установить для файла hello права на исполнение (x), ввести имя файла в командной строке (./hello) и нажать Enter
3.