Лабораторная работа. Права доступа к файлам.

Задачи

1. Просмотр и понимание разрешений на файлы и каталоги
2. Применение команды chmod для изменения разрешений
3. Применение команды chown для изменения владельца
4. Просмотр специальных разрешений

Описание

Ниже приведены команды для работы в ОС Linux в консольном режиме. Изучите приведенные команды и отработайте их в командной строке Linux. При выполнении работы проявите творчество и поэкспериментируйте с командами.

Изменение прав на доступ к файлам и директориям

Создайте два каталога и два файла в директории /tmp
root@linux-pc:~# cd /tmp
username@linux-pc:/tmp$ mkdir priv-dir pub-dir
username@linux-pc:/tmp$ touch priv-dir/priv-file
username@linux-pc:/tmp$ touch pub-dir/pub-file
username@linux-pc:/tmp$

Если вы хотите сделать каталог более закрытым, вы можете использовать команду chmod для удаления имеющихся по умолчанию разрешений. Используйте команду chmod для удаления разрешений для чтения и выполнения для глобальных разрешений.
username@linux-pc:/tmp$ ls -ld priv-dir

username@linux-pc:/tmp$ chmod o-rx priv-dir
username@linux-pc:/tmp$ ls -ld priv-dir

Используйте команду chmod для удаления любых разрешений для группы-владельца и глобальных разрешений.
username@linux-pc:/tmp$ ls -l priv-dir/priv-file

username@linux-pc:/tmp$ chmod q-rw,o-r priv-dir/priv-file
username@linux-pc:/tmp$ ls -l priv-dir/priv-file

Предоставьте всем пользователям одинаковые разрешения на чтение и запись для pub-file.
username@linux-pc:/tmp$ ls -l pub-dir/pub-file
username@linux-pc:/tmp$ chmod a=rw pub-dir/pub-file
username@linux-pc:/tmp$ ls -l pub-dir/pub-file

Создайте файл test.sh в каталоге /tmp, содержащий команду date.
username@linux-pc:/tmp$ echo "date" > test.sh

Попытайтесь выполнить (запустить) файл. Вы должны получить ошибку.
username@linux-pc:/tmp$ ./test.sh
Выведите разрешения на файл, чтобы понять причину.
username@linux-pc:/tmp$ ls -l test.sh

Установите разрешения на выполнение для файла test.sh и затем попробуйте выполнить (запустить) файл еще раз. Помните, что только владелец файла или пользователь root обладает правами на изменение разрешений для файла.
username@linux-pc:/tmp$ chmod u+x test.sh
username@linux-pc:/tmp$ ls -l test.sh

username@linux-pc:/tmp$ ./test.sh

Используя восьмеричные права измените разрешения файла test.sh так, чтобы каждый пользователь системы мог выполнить файл.
username@linux-pc:/tmp$ chmod 775 test.sh
username@linux-pc:/tmp$ ls -l test.sh

Используйте команду chown для изменения владельца и группы-владельца файла pub-dir для пользователя root и группы root. Выведите информацию о каталоге.
root@linux-pc:/tmp# chown root:root pub-dir
username@linux-pc:/tmp$ ls -ld pub-dir

Просмотр специальных разрешений

Посмотрите информацию о каталогах /tmp и /var/tmp.
username@linux-pc:/tmp$ ls -ld /tmp
username@linux-pc:/tmp$ ls -ld /var/tmp

Каталоги /tmp и /var/tmp читаются, записываются и исполняются для всех. Помимо домашних каталогов пользователей, два «временных» каталога - это места в файловой системе, где обычные пользователи могут создавать новые файлы или каталоги.
Это создает проблему: если все пользователи могут создавать новые файлы, они также могут удалять существующие файлы. Это связано с тем, что разрешение на запись в каталоге предоставляет пользователям возможность добавлять и удалять файлы в каталоге.
Символ t в столбце execute для других разрешений указывает, что этот каталог имеет набор разрешений Sticky Bit. Это специальное разрешение означает, что даже если каждый может добавлять файлы в эти каталоги, только пользователь, создавший файл, может удалить этот файл.
Просмотрите разрешения на файл /etc/shadow.
username@linux-pc:/tmp$ ls -l /etc/shadow

Когда пользователь обновляет свой пароль командой passwd, команда passwd выполняется с помощью специального разрешения, называемого setuid. Разрешение setuid приводит к тому,

что файл выполняется как пользователь, которому принадлежит файл, а не пользователь, который фактически выполняет команду.
Если вы исходите из опыта Microsoft Windows, setuid похож на функцию «Запуск от имени администратора», предоставляемую в Windows.
username@linux-pc:/tmp$ ls -l /usr/bin/passwd

Создание ссылок

Создайте текстовый файл source.
username@linux-pc:/tmp$ echo "This is source file" > source

Используйте команду ln для создания жесткой ссылки. Просмотрите информацию об источнике исходного файла и файле жесткой ссылки:
username@linux-pc:/tmp$ ln source hardlink

Сравните вывод команды ls. Жесткая ссылка создает ссылку на туже область диска, что и исходное имя файла, информация будет сохранена до тех пор, пока существует хотя бы одна жесткая ссылка.
username@linux-pc:/tmp$ ls -li source hardlink

Создайте символическую ссылку на исходный файл и просмотрите сведения о файлах.
username@linux-pc:/tmp$ ln -s source softlink
username@linux-pc:/tmp$ ls -li source softlink

Создайте символьную ссылку каталога /proc.
Успех этой команды показывает, что символьные ссылки могут ссылаться на каталоги, и они могут переходить из одной файловой системы в другую. Жесткие ссылки для этих задач использоваться не могут.
username@linux-pc:/tmp$ ln -s /proc crossdir
username@linux-pc:/tmp$ ls -l crossdir