Лабораторная работа № 4

Тема: Работа с правами доступа и атрибутами файлов

Цель: Освоение основ работы с правами доступа и атрибутами файлов в операционной системе Linux.

В Linux система прав доступа определяет, кто и в каком объеме имеет доступ к файлам и директориям. Права доступа к файлам обозначаются с использованием букв r (чтение), w (запись) и x (выполнение). Эти права разделяются между тремя классами пользователей: владельцем файла, группой файла и остальными пользователями.

1. Основы системы прав доступа:

1. **Чтение (r):**
   - *Файл:* Пользователь или группа имеют право читать содержимое файла.
   - *Директория:* Пользователь или группа имеют право просматривать список файлов в директории.
2. **Запись (w):**
   - *Файл:* Пользователь или группа имеют право изменять содержимое файла, включая создание и удаление файлов.
   - *Директория:* Пользователь или группа имеют право создавать, удалять и изменять файлы в директории.
3. **Выполнение (x):**
   - *Файл:* Пользователь или группа имеют право выполнить файл, если он является исполняемым.
   - *Директория:* Пользователь или группа имеют право входить в директорию.

Три варианта записи прав пользователя

двоичная	восьмеричная	символьная	права на файл	права на директорию
000	0	---	нет	нет
001	1	--x	выполнение	чтение файлов и их свойств
010	2	-w-	запись	нет
011	3	-wx	запись и выполнение	всё, кроме чтения списка файлов
100	4	r--	чтение	чтение имён файлов
101	5	r-x	чтение и выполнение	доступ на чтение
110	6	rw-	чтение и запись	чтение имён файлов
111	7	rwx	все права	все права

Представление прав доступа:

Права доступа к файлам представляются в виде строки из 10 символов. Первый символ обозначает тип файла (обычный файл, директория и т. д.), а следующие три группы по три символа каждая обозначают права доступа для владельца, группы и остальных пользователей.

Пример строки прав доступа: **-rwxr-xr--**

- **-** - обычный файл.
- **rwx** - права доступа для владельца (чтение, запись, выполнение).
- **r-x** - права доступа для группы (чтение, выполнение).
- **r--** - права доступа для остальных пользователей (только чтение).

Владелец - это пользователь, который создал файл или директорию.

Группа - это набор пользователей, объединенных для облегчения управления правами доступа. Владелец файла может принадлежать к определенной группе, и группа может иметь свои собственные права доступа к файлу.

Команды для работы с правами доступа:

1. **chmod:**
   - Изменение прав доступа к файлам и директориям.
   - Пример: **chmod +x file** (добавить право выполнения).
2. **chown:**
   - Изменение владельца файла или директории.
   - Пример: **chown user:group file** (изменить владельца и группу).
3. **chgrp:**
   - Изменение группы файла или директории.
   - Пример: **chgrp group file** (изменить группу).

Примеры использования:

1. **Просмотр прав доступа к файлам и директориям:**
   `ls -l`
2. **Изменение прав доступа к файлу:**
   `chmod +rwx filename`
3. **Изменение владельца файла:**
   `chown newowner filename`
4. **Изменение группы файла:**
   `chgrp newgroup filename`

2. Определение текущих прав доступа к файлам и директориям с использованием команды ls -l.

Команда **ls -l** в Linux позволяет отобразить детальную информацию о файлах и директориях в текущем рабочем каталоге. Включая права доступа, владельца, группу, размер, дату последнего изменения и другие атрибуты. Вот как можно определить текущие права доступа с использованием этой команды:

```
ls -l
```

Результат будет примерно следующим:

```
-rw-r--r-- 1 user group 1024 Dec 8 10:00 example.txt drwxr-xr-x 2 user group 4096 Dec 8 10:00 example_directory
```

В приведенном выше примере:

- **rw-r--r--** - это строка, представляющая права доступа к файлу **example.txt**. Первый символ (**-** в данном случае) указывает на тип файла (обычный файл).
- **1** - количество жестких ссылок на файл.
- **user** - владелец файла.
- **group** - группа файла.
- **1024** - размер файла в байтах.
- **Dec 8 10:00** - дата последнего изменения файла.
- **example.txt** - имя файла.

Преобразуем строку **rw-r--r--**:

- Первые три символа (**rw-**) представляют права доступа для владельца (чтение и запись).
- Следующие три символа (**r--**) представляют права доступа для группы (только чтение).
- Последние три символа (**r--**) представляют права доступа для остальных пользователей (только чтение).

В примере с директорией **example_directory**, **drwxr-xr-x** также указывает на права доступа, но буква **d** в начале означает, что это директория.

3. Изменение прав доступа

Изменение прав доступа с использованием символьного представления:

1. **Добавление прав:**
   - Для добавления прав доступа используются символы **+**. Например, чтобы добавить право выполнения для всех пользователей:

     ```
     chmod +x filename
     ```
2. **Удаление прав:**
   - Для удаления прав доступа используются символы **-**. Например, чтобы удалить право записи для группы:

     ```
     chmod g-w filename
     ```
3. **Установка конкретных прав:**
   - Чтобы установить конкретные права, используйте символ **=**. Например, чтобы установить только права чтения для владельца:

     ```
     chmod u=r filename
     ```

Изменение прав доступа с использованием числового представления:

1. **Числовое представление прав:**
   - Каждая комбинация прав представляется числом от 0 до 7:
     - **r** (чтение) = 4
     - **w** (запись) = 2
     - **x** (выполнение) = 1
   - Пример: права **rwx** = 4 + 2 + 1 = 7.
2. **Изменение прав с использованием чисел:**
   - Например, чтобы установить права **rw-r--r--** для владельца, используйте:

     ```
     chmod 644 filename
     ```
3. **Комбинирование чисел:**
   - Первая цифра для владельца, вторая для группы, третья для остальных. Например, чтобы установить **rwxr-xr--**, используйте:

     ```
     chmod 754 filename
     ```

Рекурсивное изменение прав для директорий:

Если вам нужно изменить права для всех файлов внутри директории и ее поддиректорий, используйте опцию **-R** (рекурсивно):

```
chmod -R 755 directory
```

Это изменит права для всех файлов и поддиректорий внутри указанной директории.

Примеры:

1. **Добавление права записи для группы и остальных:**
   `chmod g+w,o+w filename`
2. **Установка прав rwx для владельца, rx для группы и остальных:**
   `chmod 711 filename`
3. **Добавление права записи для всех:**
   `chmod a+w filename`
4. **Установка прав rwxr-xr-x для всех:**
   `chmod 755 filename`

4. Атрибуты SUID, SGID и Sticky Bit.

Атрибуты SUID (Set User ID), SGID (Set Group ID) и Sticky Bit - это специальные атрибуты, которые можно установить на исполняемых файлах и директориях в системах Linux. Эти атрибуты позволяют изменять стандартное поведение файлов в отношении прав доступа и безопасности. Вот их краткое описание:

1. **SUID (Set User ID):**
   - Когда установлен атрибут SUID на исполняемом файле, он выполняется с привилегиями владельца файла, а не тем, кто запускает файл.
   - Это может быть полезным, например, при выполнении программ, которые требуют привилегий пользователя root.
   - Пример: **`chmod +s file`** или **`chmod 4755 file`**.
2. **SGID (Set Group ID):**
   - Когда установлен атрибут SGID на исполняемом файле, он выполняется с привилегиями группы файла, а не группы пользователя, который его запустил.
   - Это может быть полезным, например, для обеспечения доступа к файлам и директориям в рамках определенной группы.
   - Пример: **`chmod +s file`** или **`chmod 2755 file`**.
3. **Sticky Bit:**
   - Когда установлен Sticky Bit на директории, только владелец файла имеет право удалять или изменять файлы в этой директории, даже если у других пользователей есть права записи в этой директории.
   - Обычно используется для директории **`/tmp`**, чтобы предотвратить удаление файлов другими пользователями.
   - Пример: **`chmod +t directory`** или **`chmod 1757 directory`**.

Примеры использования:

1. **SUID:**
   - Установка SUID для выполнимого файла:
     `chmod +s executable_file`
   - Проверка установки SUID:
     `ls -l`

Если атрибут SUID установлен, вы увидите букву "s" вместо буквы "x" в разряде выполнения для владельца файла. Например:
-rwsr-xr-x 1 user group 1024 Dec 8 10:00 executable_file

2. **SGID:**
   - Установка SGID для выполнимого файла:
     bashCopy code
   - chmod g+s executable_file Проверка установки SGID:
     ls -l executable_file

После выполнения этой команды, если вы посмотрите на вывод команды ls -l для файла, вы увидите букву "s" в разряде выполнения для группы файла. Например:
-rwxr-sr-x 1 user group 1024 Dec 8 10:00 executable_file

3. **Sticky Bit:**
   - Установка Sticky Bit для директории:
     chmod +t directory
   - Проверка установки Sticky Bit:
     ls -ld directory

Если Sticky Bit установлен, вы увидите букву "t" в выводе. Например:
drwxrwxrwt 2 user group 4096 Dec 8 10:00 directory

Примечание:

- Обратите внимание, что использование SUID и SGID может создавать потенциальные уязвимости безопасности, поэтому они должны использоваться осторожно и только при необходимости.
- Использование Sticky Bit на директориях может быть полезным для общедоступных директорий, таких как **/tmp**, чтобы предотвратить удаление файлов другими пользователями.

5. Использование umask

umask в Linux определяет значения прав доступа по умолчанию, которые будут отключены при создании новых файлов и директорий. Он устанавливается для каждого пользователя и определяет "маску" прав, которые не будут использоваться при создании нового файла или директории. Установка **umask** влияет на конечные права доступа файла или директории.

В **umask** используются восьмеричные числа для представления значений прав доступа, которые не будут использоваться. Например, если установить **umask 022**, это означает, что права на запись для группы и остальных пользователей будут отключены.

Примеры использования **umask**:

1. **Установка umask:**
   umask 022
2. **Создание файла:**
   touch new_file
   В результате создания файла **new_file** права доступа будут следующими: **-rw-r--r--**.
3. **Создание директории:**
   mkdir new_directory
   В результате создания директории **new_directory** права доступа будут следующими: **drwxr-xr-x**.
4. **Применение umask к постоянным настройкам пользователя:** Для того чтобы установить **umask** постоянно для пользователя, его можно добавить в файл **~/.bashrc** (или **~/.bash_profile**, **~/.profile** в зависимости от используемого оболочки).
   Например, в **~/.bashrc**:

```
umask 022
```

После изменения этого файла, новые сеансы терминала будут использовать установленное значение **umask**.

5. **Сброс umask к значению по умолчанию:**

```
umask -S
```

Эта команда покажет текущее значение **umask**. Если вы хотите вернуть его к значению по умолчанию, используйте:

```
umask 0022
```

Важно отметить, что **umask** не управляет правами выполнения (x) для файлов и директорий. Эти права должны быть установлены явно при необходимости.

6. Команда **chgrp**

Команда **chgrp** в Linux используется для изменения группы, к которой принадлежит файл или директория. Она позволяет вам изменять группу файла, не меняя его владельца (пользователя).

Синтаксис команды **chgrp** следующий:

```
bashCopy code
chgrp [опции] новая_группа файл_или_директория
```

Примеры использования:

1. **Изменение группы файла:**

```
bashCopy code
chgrp newgroup file.txt
```

Эта команда изменит группу файла **file.txt** на **newgroup**.

2. **Изменение группы директории и её содержимого (рекурсивно):**

```
bashCopy code
chgrp -R newgroup directory
```

Опция **-R** применяет изменения рекурсивно ко всем файлам и поддиректориям внутри указанной директории.

3. **Изменение группы нескольких файлов:**

```
bashCopy code
chgrp group1 file1.txt file2.txt
```

Эта команда изменит группу файлов **file1.txt** и **file2.txt** на **group1**.

4. **Изменение группы с использованием числового идентификатора группы (GID):**

```
bashCopy code
chgrp 1000 file.txt
```

В этом примере **1000** - это числовой идентификатор группы (GID), и файлу **file.txt** будет назначена группа с этим идентификатором.

5. **Изменение группы с использованием переменной окружения:**

```
bashCopy code
chgrp $USER file.txt
```

В этом примере **USER** - это переменная окружения, предоставляющая имя текущего пользователя. Файлу **file.txt** будет назначена группа, соответствующая текущему пользователю.

Задание:

1. Создайте файл, например, "document.txt".
Задайте следующие права доступа:
Владелец: чтение, запись, выполнение.
Группа: чтение.
Остальные: запрет на доступ.

2. Создайте каталог "public_data" и установите следующие права:
Владелец: чтение, запись, выполнение.
Группа: чтение, запись.
Остальные: чтение.
Измените права доступа к каталогу так, чтобы только владелец имел права на запись.

3. Создайте несколько файлов и установите следующие числовые права:
Файл 1: 744
Файл 2: 624
Файл 3: 711
Объясните их влияние на права доступа.

4. Создайте каталог "project_data" и несколько подкаталогов и файлов внутри него.
Установите различные права для каталога и его содержимого.
Используйте команду chmod с опцией -R, чтобы рекурсивно изменить права внутри "project_data".

5. Настройте umask так, чтобы создаваемые файлы имели права доступа 640, а директории 750.
Создайте новый файл и директорию и проверьте, соответствуют ли их права установленным значениям.

6. Создайте директорию "confidential_data".
Установите Sticky Bit на этот каталог, чтобы только владелец мог удалять свои файлы внутри него. Смените пользователя и проверьте, что правило работает.

7. Создайте файл и установите произвольные права.
С помощью команд chown и chgrp измените владельца и группу файла.

Отчет должен содержать:

1. Заголовок (название лабораторной работы, название группы, фамилию студента);
2. Скриншоты, подтверждающие выполнение заданий;
3. Выводы.