**Redireccionamiento de Entrada/Salida en Linux:**

En sistemas operativos como Linux y Unix, cada operación dentro de un proceso implica una entrada y una salida (I/O), y a veces, una salida de error. Para facilitar la manipulación de datos, estos sistemas permiten redireccionar la entrada y salida de los comandos.

* **Entrada estándar y salida estándar:**
  + La entrada estándar generalmente es el teclado, mientras que la salida estándar es la ventana o terminal actual.
  + Los comandos siguen un esquema básico:
    - Si no hay redireccionamiento, la entrada y salida son estándar.
    - Si ocurre un error, la salida es la salida estándar de error.
* **Operadores de redireccionamiento:**
  + **>**: Redirecciona la salida a un archivo. Si el archivo existe, sobrescribe.
  + **>>**: Redirecciona la salida al final de un archivo (append). Si el archivo no existe, lo crea.
  + **<**: Redirecciona la entrada del comando desde un archivo.
* **Ejemplos:**

cat > archivo1 # Entrada estándar desde el teclado, salida al archivo archivo1

ls /etc >> lista # Lista el directorio /etc y redirige la salida al archivo lista

**En un directorio con permiso de escritura se puede:** (crear nuevos archivos y directorios o borrarlos).

• Añadir y borrar ficheros, aunque sobre estos no se tengan permisos de escritura.

• Añadir directorios y borrarlos si los permisos de estos lo permiten. Es decir si tenemos más directorios debajo de este, dependerá de los permisos que tengan esos otros directorios.

• Modificar ficheros siempre que los permisos de estos lo permitan.

**En un directorio sin permiso de de escritura se puede:**

• No se puede añadir ni borrar ficheros ni directorios.

• Se puede modificar el contenido de los ficheros siempre que se tenga permiso de escritura sobre ellos

**En un directorio con permiso de lectura se puede:**

• Se puede ver lo que hay dentro del directorio con ls.

**En un directorio sin permiso de lectura:**

• No se puede ver lo que hay dentro. El contenido del directorio no es visible ( ls ).

**En un directorio sin permiso de ejecución.**

• No se puede hacer nada, porque nos impide utilizar su nombre en una trayectoria y no podemos acceder a él de ninguna manera. Es la mejor forma de proteger un directorio. A este permiso se le suele llamar permiso de paso. Significa que no podemos entrar en la carpeta con el (comando cd).

**Seguridad de Archivos y Directorios:**

En sistemas Linux, la seguridad de archivos y directorios se basa en permisos y propietarios.

* **Propietario y Grupo Propietario:**
  + El usuario propietario es quien crea el archivo o directorio.
  + El grupo propietario : es el grupo del usuario creador.
* **Permiso de Acceso:**
  + Para archivos: lectura, escritura y ejecución.
  + Para directorios: listar, crear/borrar archivos y directorios, buscar.
  + Los permisos afectan al usuario propietario, grupo propietario y otros usuarios.
* **Manipulación de Permisos:**
  + Se pueden manipular con el comando **chmod**:
    - Notación octal: **chmod 755 archivo**
    - Notación simbólica: **chmod u+w archivo**

**Explicación de Comandos:**

Se explican los comandos **chmod** con ejemplos:

* **chmod o+r carta**: Agrega permiso de lectura para otros usuarios.
* **sudo chmod 755 /bin**: Asigna todos los permisos al propietario y permisos de lectura y ejecución a grupo y otros usuarios.

**Discusión Adicional:**

La notación octal es una forma de representar los permisos de archivos y directorios en sistemas Unix y Linux. Utiliza números en base 8 para codificar los permisos de lectura, escritura y ejecución, asignando un valor específico a cada combinación de permisos.

En la notación octal, cada permiso se representa mediante un número:

* Lectura (r): 4
* Escritura (w): 2
* Ejecución (x): 1

Para calcular los permisos octales de un archivo o directorio, se suman los valores correspondientes a los permisos que se desean otorgar. Por ejemplo:

* Si se otorgan permisos de lectura y escritura, se suma 4 (lectura) y 2 (escritura), lo que da como resultado 6.
* Si se otorgan permisos de lectura, escritura y ejecución, se suma 4 (lectura), 2 (escritura) y 1 (ejecución), lo que da como resultado 7.

Aquí hay un resumen de los valores octales para diferentes combinaciones de permisos:

* Ningún permiso: 0
* Ejecución: 1
* Escritura: 2
* Escritura y ejecución: 3
* Lectura: 4
* Lectura y ejecución: 5
* Lectura y escritura: 6
* Todos los permisos (lectura, escritura y ejecución): 7

Por ejemplo, si queremos otorgar permisos de lectura y escritura al propietario y solo permisos de lectura al grupo y a otros usuarios, podríamos usar el valor octal 664:

* Propietario: Lectura (4) + Escritura (2) = 6
* Grupo: Lectura (4)
* Otros: Lectura (4)

Por lo tanto, el valor octal para estos permisos sería 664. Esta notación proporciona una forma concisa de representar los permisos de archivos y directorios en sistemas Unix y Linux.

Ejemplos

* + - **chmod 755 archivo ( permisos usuario rwx grupo r-x otros r-x)**

La notación simbólica es otra forma de representar los permisos de archivos y directorios en sistemas Unix y Linux. Utiliza una serie de letras y símbolos para denotar los diferentes tipos de permisos y a quién se aplican. Aquí está la estructura básica de la notación simbólica:

* **u**: Representa al propietario del archivo.
* **g**: Representa al grupo al que pertenece el archivo.
* **o**: Representa a otros usuarios que no son el propietario ni pertenecen al grupo.
* **a**: Representa a todos los usuarios (equivalente a "ugo" juntos).

Luego, se utilizan los siguientes símbolos para denotar los permisos:

* **+**: Agrega permisos.
* **-**: Quita permisos.
* **=**: Establece permisos exactamente como se especifica, reemplazando los permisos anteriores.

Y finalmente, se usan los siguientes símbolos para representar los permisos específicos:

**r**: Lectura (read).

**w**: Escritura (write)

.**x**: Ejecución (execute).

Por ejemplo, si deseamos otorgar permisos de lectura y escritura al propietario y solo permisos de lectura al grupo y a otros usuarios, podemos usar la notación simbólica de la siguiente manera:

chmod u=rw,g=r,o=r archivo.txt

Esto significa:

* "u=rw": Establece los permisos de lectura y escritura para el propietario.
* "g=r": Establece el permiso de solo lectura para el grupo.
* "o=r": Establece el permiso de solo lectura para otros usuarios.

La notación simbólica proporciona una forma más legible y fácil de entender para asignar y modificar los permisos de archivos y directorios en comparación con la notación octal.

**Ejemplos con a- y a+**

1. **Permiso de lectura, escritura y ejecución para el propietario, grupo y otros:**

chmod a+rwx archivo.txt

Esto otorga permisos de lectura, escritura y ejecución para el propietario, el grupo y otros usuarios en el archivo "archivo.txt".

1. **Quitar permiso de ejecución para otros:**

chmod a-x archivo.sh