# OBJETIVOS

* + Aprender a utilizar la redirección de comandos, manejo de errores y tuberías.

# SEGURIDAD

|  |  |
| --- | --- |
| **220px-DIN_4844-2_Warnung_vor_einer_Gefahrenstelle_D-W000.svg.png** | **Advertencia:**  En este laboratorio está prohibida la manipulación de hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas. |

# NORMAS EMPLEADAS

* + Ubicar maletines y/o mochilas en el gabinete al final de aula de laboratorio o en los casilleros asignados al estudiante.
  + Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

# RECURSOS

* + Computadora con Vmware Workstation.
  + Máquina virtual con CentOS 7.

# METODOLOGÍA PARA EL DESARROLLO DE LA TAREA

* + El laboratorio se desarrolla de forma individual.

# PROCEDIMIENTO

* Se empleará una máquina virtual con Ubuntu Server 18.04.
* El hostname del sistema operativo deberá contar con la siguiente estructura: **primera letra de su primer nombre, seguido de su apellido paterno** y de las letras **“-srv”**. Un nombre válido sería **jdiaz-srv** para alguien que se llame Juan Diaz.

***NOTA***

Toda captura de pantalla deberá ir acompaña del hostname de sistema operativo.

* Ingrese con su usuario principal.

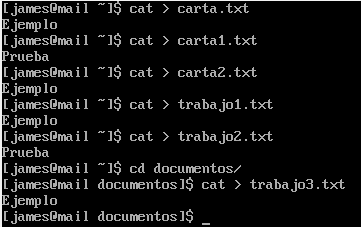
## REDIRECCIÓN DE COMANDOS

* Crear una cuenta de usuario para: Jaime Ames

|  |
| --- |
| $ sudo useradd -m -s /bin/bash james  $ sudo passwd james |

* Inicie una sesión en una consola o terminal con la cuenta de james.  
  
* Crear los siguientes directorios dentro del directorio hogar de james.
  + ventas
  + informes
  + documentos
  + documentos/privados/  
      
    
* Crear los siguientes archivos, con los contenidos indicados:

|  |  |
| --- | --- |
| **Archivo** | **Contenido** |
| /home/james/carta | Ejemplo |
| /home/james/carta1 | Prueba |
| /home/james/carta2 | Ejemplo |
| /home/james/trabajo1 | Ejemplo |
| /home/james/trabajo2 | Prueba |
| /home/james/documentos/trabajo3 | Ejemplo |

*   
  Como usuario **james** y estando en el directorio hogar de **james** muestre el contenido de este directorio, indicar el comando utilizado.

|  |
| --- |
|  |

* Liste solamente los archivos y NO los directorios, envíe el resultado a un archivo llamado **lista.txt**. Indicar el comando utilizado.

|  |
| --- |
|  |

* ¿Cuál es el contenido del archivo **lista.txt**?

|  |
| --- |
|  |

* Liste los archivos que empiecen con **trabaj** y envíe la salida del comando al mismo archivo anterior:

|  |
| --- |
|  |

* Visualice el contenido del archivo:

|  |
| --- |
|  |

* ¿Qué pasó con el contenido anterior?

|  |
| --- |
| **El contenido anterior se sobrescribió.** |

* Liste los archivos que empiecen con **carta** y añada la salida de este comando al archivo anterior.

|  |
| --- |
| $ ls carta\* >> lista.txt |

* ¿Qué pasó con el contenido anterior?

|  |
| --- |
| **Esta vez, se conservó el contenido anterior, y se añadió el nuevo contenido.** |

* Hemos utilizado la salida del comando **ls** para redireccionarla a un archivo, también podemos utilizar otros comandos, pruebe los siguientes comandos y visualice el contenido de los archivos resultantes:

|  |
| --- |
| $ w > salida1.txt  $ find /etc –name motd > salida2.txt |

* ¿Qué pasó en el primer comando?

|  |
| --- |
| **En este caso, w devuelve una lista de todos los usuarios con sesión abierta y las terminales en uso.** |

* ¿Qué pasó en el segundo comando?

|  |
| --- |
| **Linux nos deniega el permiso al ejecutar estos comandos.** |

***NOTA***

También se puede utilizar los comandos de redirección para inicializar un archivo (archivo vacío). Esto es útil para aplicativos que utilizan, por ejemplo, ciertos archivos para guardar mensajes de error los cuales crecen periódicamente, pero no pueden ser eliminados porque los requiere el aplicativo.

* Por ejemplo supongamos que Usted necesita guardar un histórico de los usuarios conectados al sistema en un archivo llamado **usuarios.txt** periódicamente ¿Cuál sería el comando a ejecutar?

|  |
| --- |
| **W >> usuarios.txt** |

* Ejecute el comando anterior varias veces. Verifique el tamaño del archivo indicando el tamaño del mismo.

|  |
| --- |
| $ ls –l usuarios.txt    El archivo usuarios.txt tiene un peso de 1596 KB |

* Como notará este archivo va creciendo en tamaño. Supongamos que Usted quiere ahora inicializar este archivo (no eliminarlo), entonces ejecute el siguiente comando:

|  |
| --- |
| $ > usuarios.txt |

* Verifique el tamaño del archivo. ¿Cuál es su tamaño?

|  |
| --- |
| **El archivo tiene un peso de 0 KB** |

* Con ayuda del comando **find**, busque un archivo cuyo tamaño esté entre 20 y 40 megas, luego copie dicho archivo al directorio hogar del usuario **james**.

|  |
| --- |
| **Primero, se creó un archivo de 36 MB.   Luego se procede a realizar la búsqueda.**  **Intentando copiar el archivo a /home**    **ME dice “permiso denegado”** |

* Ejecute el siguiente comando:

|  |
| --- |
| $ split **<archivo copiado>** -b 4MB |

* Liste los archivos en el directorio hogar que empiezan con las letras **xa**

|  |
| --- |
|  |

* ¿Qué sucede si ejecuta el siguiente comando?

|  |
| --- |
| $ split **<archivo copiado>** -b 4MB part\_ |

|  |
| --- |
|  |

* ¿Qué sucede si ejecuta el siguiente comando?

|  |
| --- |
| $ cat part\_\* > arcuno |

|  |
| --- |
|  |

* Hemos visto cómo se puede redireccionar la salida de un comando con el carácter mayor (**>**). También se puede redireccionar la entrada de un comando con el carácter menor **(<)**. Para ilustrar ésto vamos a utilizar el comando **write**.
* Utilizando su usuario principal, crear una cuenta de usuario para: Francisco García

|  |
| --- |
| $ sudo useradd -m -s /bin/bash fgarcia  $ sudo passwd fgarcia |

* Inicie una sesión en una consola o terminal con la cuenta de **fgarcia**. Verifique quiénes están conectados:

|  |
| --- |
|  |

* En la consola de texto del usuario **james**, cree un archivo llamado **mensaje** utilizando el editor de texto VI, su contenido será el siguiente:

**Este es un archivo de prueba**

**de cómo se utiliza el carácter**

**de redirección de entrada**

* Utilice el comando **write** para enviar el contenido del archivo mensaje al terminal de un usuario.

|  |
| --- |
| $ write fgarcia < mensaje |

* Verifique que el mensaje haya llegado a la consola de texto donde ingresó con el usuario **fgarcia**.

|  |
| --- |
|  |

* El comando **ps –aux** muestra una lista de procesos que están ejecutándose en nuestro servidor. Utilice el comando **tee** para guardar la salida estándar del comando **ps –aux** al archivo llamado **procesos.txt**

|  |
| --- |
| **…**  **…**  **…** |

* Se desea que al final del archivo **procesos.txt** se pueda ver la lista de usuarios conectados y nuevamente los procesos en ejecución. ¿Cuál sería el comando?

|  |
| --- |
|  |

## MANEJO DE ERRORES

* Ingrese a una sesión con su usuario principal y cree un directorio dentro de **/home** con acceso restringido.

|  |
| --- |
| $ cd /home  $ sudo mkdir prueba  $ chmod ***XYZ*** prueba |

***Nota***: XYZ son valores numéricos que tendrán que reemplazarse.

* Los errores que suceden al ejecutar un comando también se pueden redireccionar. En la consola de texto en la que ingresó el usuario **james**, ejecute los siguientes comandos:

|  |
| --- |
| $ ls anual |

* ¿Le muestra mensajes de error? ¿A qué se debe esto?

|  |
| --- |
| **Sí, porque el fichero no existe** |

|  |
| --- |
| $ find /home –name hola |

* ¿Le muestra mensajes de error? ¿A qué se debe esto?

|  |
| --- |
| **Sí, porque el usuario no cuenta con los respectivos permisos.** |

* Redireccione los mensaje de error anteriores a un archivo, para ello utilice el carácter de redirección **2>**

|  |
| --- |
|  |

* ¿Qué información es la que se muestra?

|  |
| --- |
| **Se muestra el mensaje de error: No existe el fichero o directorio** |

* Verifique el contenido de los archivos:

|  |
| --- |
| $ cat error-ls  $ cat error-find |

* ¿Qué contienen estos archivo?

|  |
| --- |
| **Contienen mensajes de error** |

* Redireccione ahora también la salida del comando

|  |
| --- |
| $ find /home -name mensaje 1>resul2.txt 2>error2.txt |

* Visualice el contenido de estos archivos. ¿Qué contiene el archivo **resul2.txt**? ¿Qué contiene el archivo **error2.txt**? En ambos casos fundamente su respuesta

|  |
| --- |
|  |

* Redireccione la salida del comando y los errores a un mismo archivo. Pruebe el siguiente comando:

|  |
| --- |
| $ find /home -name mensaje 1>resul3.txt 2>&1 |

* Visualice el contenido de este archivo. ¿Qué contiene el archivo resul3.txt?

|  |
| --- |
| **Contiene los mensajes de error de la búsqueda de find…** |

* Liste el contenido del directorio y redirecciónelo primero a un archivo llamado **lista2.txt**.

|  |
| --- |
|  |

* Cuente la cantidad de líneas del archivo. ¿Cómo lo puede probar en un editor VI? Utilice man para ver el funcionamiento del comando **wc**.

|  |
| --- |
| **Para contar la cantidad de lineas en vi se usó :set number** |

|  |
| --- |
|  |

## MANEJO DE TUBERÍAS (PIPES)

* Un meta-carácter muy empleado en el Sistema Operativo Linux es la tubería (**|**), este símbolo permite redireccionar la salida de un comando como entrada de otro. Por ejemplo si Usted desea ver el contenido de un directorio con pausa puede utilizar el comando:

|  |
| --- |
| $ ls /etc | more |

* Muestre la lista de usuarios conectados en forma ordenada:

|  |
| --- |
| $ who | sort |

* Si son demasiados los usuarios conectados necesitaría mostrarlo con pausa. ¿Cómo lo haría? Indique el comando a ejecutar.

|  |
| --- |
| **Se usaría ->** |

* Vamos a utilizar la tubería para crear otro comando. Supongamos que queremos ver los archivos de un directorio que corresponden a un mes determinado. Para hacer el siguiente ejemplo vamos a crear archivos con diferente fecha. Ingrese a una sesión con su usuario principal y utilizando el comando **date** o **timedatectl** cambie la fecha del sistema (el mes debe corresponde a un mes anterior):

|  |
| --- |
|  |

* En una sesión como **james**, cree algunos archivos con esta nueva fecha en su directorio hogar y pruebe utilizar el siguiente comando:

|  |
| --- |
| $ ls –l | grep ago |

* En una sesión con el usuario principal, retorne la fecha del sistema a su estado inicial.

|  |
| --- |
|  |

* También podría utilizar la tubería para extraer algunas líneas de algún comando, esto lo puede realizar utilizando el comando **head** o **tail** según corresponda. Por ejemplo el comando **w** le muestra en la primera línea el número de usuarios conectados entonces Usted podría enviar sólo esta línea a un archivo:

|  |
| --- |
| $ w | head –1 > usuarios |

* Puede ser que Usted desea una estadística de número de usuarios conectados y mantener el historial. ¿Qué comando utilizaría para realizar ésto?

|  |
| --- |
| **Usaría el comando $last** |

* Como Usted habrá notado solo figura la hora pero NO la fecha. Se podría usar este comando para resolver este problema (note el uso del punto y coma para escribir dos comando seguidos en una misma línea). Visualice el contenido del archivo usuarios:

|  |
| --- |
| $ date >> usuarios ; w | head –1 >> usuarios |

* ¿Qué hace este comando?

|  |
| --- |
| **Guarda la fecha en que los usuarios iniciaron sesión en el Sistema** |

* También con la tubería puede extraer columnas de un archivo, utilizando el comando **cut**. Por ejemplo en el comando **date** se encuentra el día de la semana el cual puede ser extraído con el siguiente comando:

|  |
| --- |
| $ date | cut --delimiter=' ' -f1 |

* A partir de todo lo anterior, proponga 02 ejemplos donde use, la redirección y las tuberías en un mismo comando. Explique los comandos propuestos. Muestre el resultado de la ejecución de los mismos. Se tomará en cuenta para la calificación la complejidad del comando (no se puede utilizar ejemplos desarrollados en clase o en el laboratorio).
* Ejemplo 1.

|  |
| --- |
|  |

* Ejemplo 2.

|  |
| --- |
|  |

# CONCLUSIONES

* Las redirecciones y tuberías pueden ser usadas de manera conjunta, con esto se obtendrá consultas más complejas.
* Una redirección consiste en trasladar datos de una entrada estándar a una salida estándar y/o viceversa.
* Por otro lado, las tuberías permiten ingresar un elemento de entrada, el cual podrá ser conectado a través de varios comandos o procesos de manera secuencial, antes de obtenerse la salida final.
* Cuando queremos guardar la salida de un comando en un archivo utilizamos el operador de redirección simple >. Si deseamos agregar más contenido, y no queremos que este se sobrescriba, usamos el operador de redirección doble >>.
* En el entorno laboral, el uso de redireccionamiento y tuberías es muy ventajoso, para el manejo de errores y el intercambio de información.

# REFERENCIAS

* <https://www.linuxtotal.com.mx/index.php?cont=redireccionamiento-en-linux>
* <https://lamiradadelreplicante.com/2011/11/10/en-las-tuberias-de-linux/>
* <https://hipertextual.com/archivo/2014/07/redirecciones-y-tuberias-bash/>