UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)

Formato para prácticas de laboratorio

PROGRAMA EDUCATIVO	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-2	12124	Taller de Sistema Operativo Unix

PRÁCTICA No.	LABORATORIO DE	Taller de Sistema Operativo Unix	DURACIÓN (HORAS)
10	NOMBRE DE LA PRÁCTICA	Redireccionamiento de Entrada	2

1. INTRODUCCIÓN

Como se mencionó en la práctica anterior, muchos mandos UNIX toman sus valores de entrada de algo conocido como entrada estándar, también conocida como stdin o standard input y envían su salida o resultados a la salida estándar también conocida como stdout o standard output. En esta práctica trabajaremos con los mandos que nos permiten modificar la entrada estándar de los mandos o lo que comúnmente se conoce como Redireccionamiento de Entrada.

2. OBJETIVO (COMPETENCIA)

El alumno conocerá y comprenderá el concepto de redireccionamiento de entrada y lo aplicará de forma eficiente en algunos mandos LINUX con actitud analitica.

3. FUNDAMENTO

Redireccionamiento de entrada.

Muchos mandos en linux necesitan para su funcionamiento recibir datos de la entrada de datos estándar. La entrada estándar recibe los datos de un dispositivo o de un archivo. El teclado representa la entrada de datos estándar por default, por lo tanto, los caracteres que son introducidos desde el teclado son colocados en la entrada estándar y posteriormente dirigidos al mando de Linux utilizado.

De manera similar a lo que sucede con la salida estándar, también se puede redireccionar la entrada estándar para recibir los datos desde un archivo en lugar que sea desde el teclado. El operador para redirigir la entrada estándar es el carácter Menor Que "<". Este proceso no produce cambios en el archivo de entrada.

Sintaxis:

mando < nombre_archivo

Ejemplo 1:

$more < carta.txt

Normalmente cuando ejecutamos el mando ***cat***, éste lee la información desde el archivo que se pasa como parámetro y muestra el contenido del archivo en pantalla. Cuando la longitud del archivo es muy extensa, la información no puede visualizarse en una sola pantalla, por lo tanto los primeros registros se van recorriendo para dar cabida a los nuevos registros que se mostrarán. Para poder visualizar la información del archivo en forma pausada, de manera que cada vez que se llene la pantalla se detenga, podemos utilizar el mando ***more*** y redireccionar su entrada como se muestra en el ejemplo 1, de forma que el archivo que se utilice como entrada sea visualizado de forma paginada.

Ejemplo 2:

El mando ***sort*** permite ordenar una lista de forma numérica o alfabética. Por ejemplo si utilizamos el mando ***sort*** para ordenar una lista de útiles escolares como se muestra a continuación:

```
$sort
 lapices
 plumas
 cuadernos
 borradores
 libros
 ^D (Oprimir Ctrl+d)
```

El mando tomará los datos del dispositivo de entrada estándar (teclado), esto es, cada uno de los útiles que estamos ingresando y mostraría en el dispositivo de salida estándar como resultado la lista anterior ordenada alfabéticamente como se muestra a continuación:

```
borradores
cuadernos
lapices
libros
plumas
```

Para el ejemplo anterior, podríamos utilizar redireccionamiento de entrada. Supongamos que los datos que tecleamos como entrada al mando ***sort*** los tenemos en un archivo llamado **listaUtiles**, entonces podríamos ordenar la lista de útiles redireccionando la entrada de los datos y obtener los mismos resultados en el dispositivo estándar de salida(pantalla) de la siguiente manera:

$sort < listaUtiles

Podríamos así como redireccionamos la entrada estándar, redireccionar la salida, de forma que la lista de útiles ordenada se envíe a un archivo en lugar de la pantalla que es el dispositivo estándar de salida como se muestra en el siguiente ejemplo:

$sort < listaUtiles > listaUtilesOrdenada

Canalización o Entubamiento.

En muchas ocasiones se tendrá la necesidad de enviar datos generados como salida por un mando a otro mando y no a un archivo destino; es decir, necesitará enviar la salida estándar de un mando hacia la entrada estándar de otro. La canalización o entubamiento es la técnica que nos permite desviar o encadenar la salida o resultado de una mando a la entrada de otro para poder realizar funciones más complejas.

El símbolo que se utiliza para efectuar el entubamiento o canalización se denomina **"pipe"** y su símbolo es **"|"**. Cuando empleamos el carácter **"|"** estamos avisando al shell de que internamente cree un mecanismo que permita la comunicación entre las dos órdenes, situadas a los lados del carácter de entubamiento. De esta manera, el shell redirige la salida de la primera al canal de comunicación, y utiliza la salida de este canal como entrada para el siguiente mando.

Sintaxis:

mando1* | *mando2 *[... y podríamos seguir agregando* ***| mando3****, etc]*

Ejemplo 3:

El mando **"*dmesg*"** se utiliza para revisar los mensajes que produce nuestro Kernel durante el arranque. Como suele ser una cantidad muy grande, podemos entubarlo al mando **"*less*"**, que nos permite ver el resultado línea a línea por medio de las flechas, o página a página presionando la barra espaciadora (y salir presionando q). Para ello interconectamos ambos mandos por medio de entubamiento.

$dmesg | less

Ejemplo 4:

Supongamos que queremos saber el número de personas que están conectadas al sistema en un instante determinado. Una forma muy sencilla de hacerlo sin utilizar entubamiento sería:

```
$ who  > usuariosConectados
$ wc -l < usuariosConectados
$ rm archivo_temporal
```

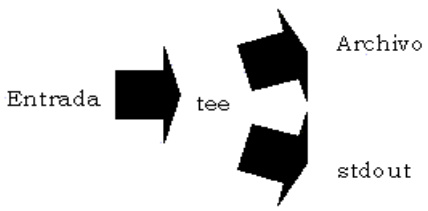
Una forma mas practica de resolver lo anterior utilizando tuberías es la siguiente:

```
$ who | wc -l
```

En el ejemplo anterior, la salida del mando ***who*** (que va a la salida estándar) se utiliza como entrada para el mando de la derecha ***wc -l*** (que lee de la entrada estándar).

Bifurcación. (*tee*)

En ciertas ocasiones nos interesa, además de redireccionar la salida de un mando a un archivo, visualizar los resultados obtenidos. El mando ***tee*** se creó con esta intención, este mando se usa para bifurcar la tubería, lee en la entrada estándar por defecto y escribe su salida en la salida estándar y en el archivo especificado. Si el archivo no existe lo crea, si existe lo sobreescribe. Si se utiliza la opción -a (append), ***tee*** añade su salida al archivo en vez de sobreescribirlo.



Sintaxis:

tee [-a] archivo

Ejemplo 5:

Supongamos que queremos tener en pantalla un listado de los procesos que esta ejecutando el usuario y almacenar el listado en un archivo llamado **procesos.txt**, hacemos lo siguiente:

$ ps | tee procesos.txt

Si ejecutamos otro mando y utilizamos ***tee*** para enviar dicha salida al mismo archivo **procesos.txt**, este será sobreescrito con la nueva salida. Si queremos que en vez de ser sobreescrito el contenido sea añadido a dicho archivo lo haremos con la opción ***-a***.

$ ls -l | tee -a procesos.txt

En el ejemplo anterior, obtendremos un listado de los archivos en formato largo que se mostrará en pantalla y simultáneamente se agregara al archivo **procesos.txt.** Si visualizamos el contenido del archivo **procesos.txt** tendremos los resultados obtenidos por los dos mandos anteriores.

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO	MATERIAL DE APOYO
Computadoras con Linux instalado	Práctica impresa

B) DESARROLLO DE LA PRÁCTICA

1. Muestra el contenido del directorio **/etc** en formato largo.
2. Muestra en forma paginada el contenido del directorio **/etc** utilizando entubamiento.
3. Genere un archivo llamado **conectados** con todos los usuarios que están conectados al sistema actualmente y simultáneamente que se visualice en pantalla.
4. Genere un archivo llamado **listado** con todos los archivos contenidos en su directorio **home** en formato largo, ordenado ascendentemente por su tamaño. El listado de archivos debe verse simultáneamente en pantalla al momento de ser generado.
5. La cuarta columna del archivo **/etc/passwd** representa el grupo al que pertenece cada uno de los usuarios del sistema, este valor puede estar repetido debido a que varios usuarios pueden pertenecer a un mismo grupo. Genere un archivo llamado **gruposSistema** que contenga una lista de todos los grupos del sistema ordenados numéricamente en forma ascendentemente y sin que se repitan los valores. El listado de grupos que tendrá el archivo deberá verse simultáneamente en pantalla al momento de ser generado.
6. Genere un archivo llamado **nombresD** que contenga el nombre de usuario (user name) del archivo **passwd** del directorio **/etc** ordenados descendentemente y simultáneamente que se visualice en pantalla.
7. Muestre el contenido del archivo **nombresD** en forma paginada sin utilizar entubamiento.

8. Utilizando el archivo **nombresD**, genera un archivo llamado **nombresA** ordenado ascendentemente y que simultáneamente se visualice en pantalla.
9. Utilizando en mando ***tee***, agrega el contenido del archivo llamado **conectados** al final del archivo **nombresD**.
10. Verifica que el archivo **nombresD** contiene los nombres en forma descendente e inmediatamente después los usuarios conectados que se agregaron en el paso anterior.
11. Ahora agrega el contenido del archivo llamado **nombresA** al final del archivo **nombresD** y que simultáneamente se vea en pantalla el contenido del primer archivo que se está agregando.
12. Verifica que el archivo **nombresD** contiene los nombres en forma descendente, inmediatamente después los usuarios conectados y finalmente el listado de los nombres de forma ascendente que se agregaron en el paso anterior.

C) CÁLCULOS (SI APLICA) Y REPORTE

No Aplica

5. RESULTADOS Y CONCLUSIONES

No Aplica

6. ANEXOS

No Aplica

7. REFERENCIAS

1. Ubuntu Manpage http://manpages.ubuntu.com/manpages/precise/en/man1/tee.1.html