

**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)
9	NOMBRE DE LA PRÁCTICA	Redireccionamiento de flujos de salida	2

1. INTRODUCCIÓN

Generalmente los programas en Unix tienen tres flujos abiertos. Estos se abren al comenzar el programa, uno es para la salida, otro para la entrada y uno es para mostrar mensajes de error o de diagnóstico. Estos flujos están atados con la terminal del usuario, aunque algunas veces se refieren a archivos o algún otro dispositivo.

El flujo de entrada también se le conoce como la entrada estándar o stdin. A el flujo de salida también se le conoce como la salida estándar o stdout y al error estándar stderr.

Formuló María Luisa González Ramírez Linda Arredondo Acosta	Revisó Aglay González Pacheco	Aprobó	Autorizó David I. Rosas Almeida
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director / Representante de la Dirección

Código: GC-N4-017 Revisión: 4

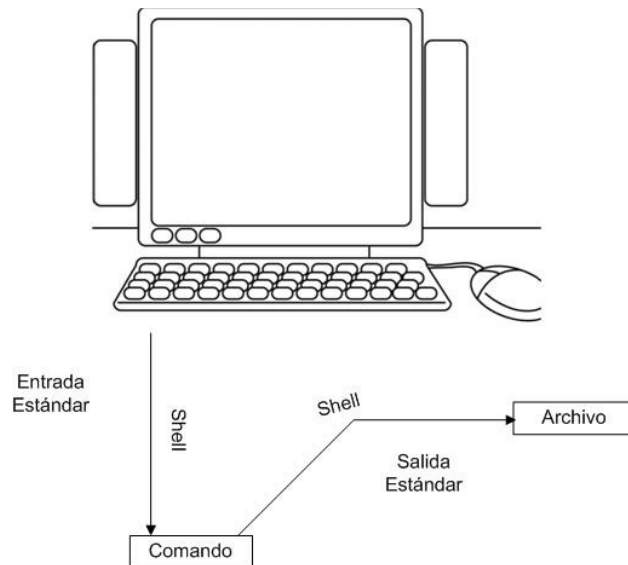


Figura 1: Flujos estándar de entrada y de salida

Estos 3 flujos se identifican con diferentes valores que se conocen como descriptor de archivos (file descriptor).

Flujo de datos	Valor
Entrada Estándar	0
Salida Estándar	1
Error Estándar	2

Estos valores son útiles al momento de redireccionar, como se mostrará más adelante.

2. OBJETIVO (COMPETENCIA)

El alumno aprenderá el uso del direccionamiento de salida, realizando ejercicios e involucrando mandos sobre la visualización de archivos con responsabilidad y perseverancia.

3. FUNDAMENTO

Redireccionamiento.

La redirección se refiere al hecho de que se puede indicar al intérprete de comandos (shell) que modifique la entrada estándar de un comando y que la salida estándar se modifique también. Por defecto el Shell asocia la entrada y la salida estándar con el teclado y el monitor. Se puede asociar la entrada o salida estándar con un comando o un archivo no solamente al monitor o al teclado.

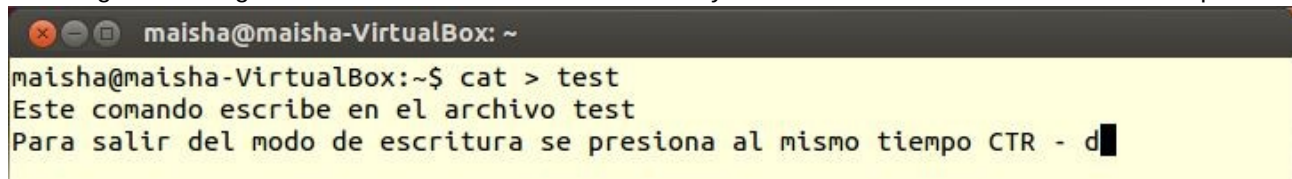
Redireccionar la salida estándar.

El símbolo de redirección de salida **n>** le dice al Shell que la salida la modifique a un archivo en lugar de al monitor. La sintaxis general es:

mando [argumentos] n> nombrearchivo

En donde el **mando** es la instrucción o el programa del cual obtendremos la salida, los **[argumentos]** son los argumentos opcionales del comando, **n** es el descriptor de archivo del flujo que se redireccionará y el **nombrearchivo** es el nombre del archivo. En caso de que no se especifique el descriptor de archivos por defecto es el 1.

En la siguiente imagen se muestra el uso del comando **cat** y el redireccionamiento de salida con el operador **>**

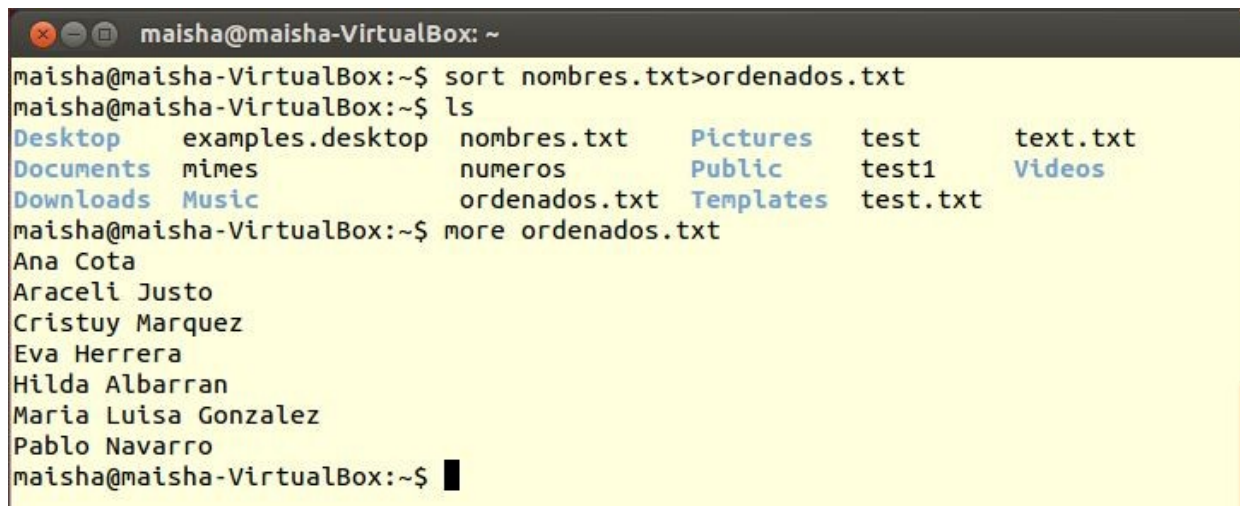


```
maisha@maisha-VirtualBox: ~  
maisha@maisha-VirtualBox:~$ cat > test  
Este comando escribe en el archivo test  
Para salir del modo de escritura se presiona al mismo tiempo CTR - d
```

Figura 2: Uso del redireccionamiento de salida

Como se observa en la figura no se especifica el descriptor de archivo por defecto es el de salida.

En la siguiente figura se muestra el resultado de utilizar el mando **sort** sobre el archivo **nombres** y se redirecciona la salida hacia un archivo con nombre **ordenados.txt**, se ejecuta el mando **ls** para ver que efectivamente se generó el archivo **ordenados.txt** y por último con el mando **more** se visualiza el contenido del archivo **ordenados.txt**



```
maisha@maisha-VirtualBox: ~  
maisha@maisha-VirtualBox:~$ sort nombres.txt>ordenados.txt  
maisha@maisha-VirtualBox:~$ ls  
Desktop      examples.desktop  nombres.txt      Pictures      test           text.txt  
Documents    mimes            numeros          Public        test1          Videos  
Downloads    Music            ordenados.txt    Templates    test.txt  
maisha@maisha-VirtualBox:~$ more ordenados.txt  
Ana Cota  
Araceli Justo  
Cristuy Marquez  
Eva Herrera  
Hilda Albarran  
Maria Luisa Gonzalez  
Pablo Navarro  
maisha@maisha-VirtualBox:~$
```

Figura 3: Redireccionamiento de salida usando el mando **sort**

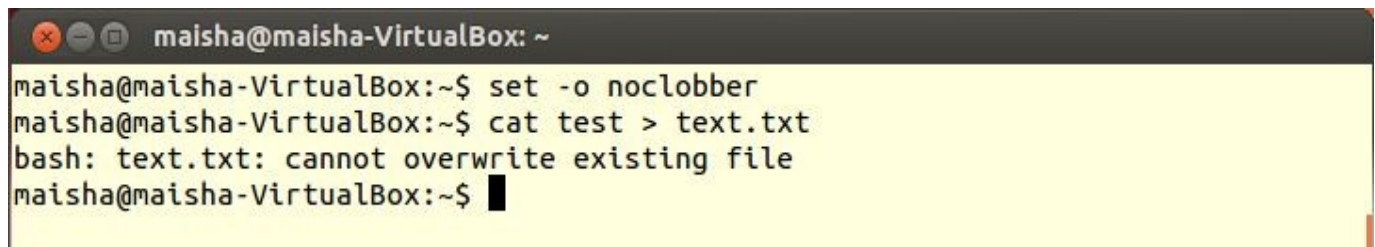
Debemos tener cuidado cuando redireccionamos la salida de un mando ya que podemos destruir un archivo que ya exista.

Para prevenir la sobreescritura de archivos utilizando la redireccionamiento de salida se modifica la característica **noclobber** por medio del comando **set**. Por ejemplo:

set -o noclobber establece *noclobber encendido*

set +o noclobber establece *noclobber apagado*

En la siguiente figura se muestra **noclobber** encendido y que muestra en la consola cuando se trata de sobrecribir un archivo por otro



```
maisha@maisha-VirtualBox: ~  
maisha@maisha-VirtualBox:~$ set -o noclobber  
maisha@maisha-VirtualBox:~$ cat test > text.txt  
bash: text.txt: cannot overwrite existing file  
maisha@maisha-VirtualBox:~$
```

Redireccionamiento del error estándar.

Podemos hacer que los errores que ocurren al ejecutar un comando se envíen a un archivo o simplemente que no los tomamos en cuenta.

Para enviar los errores a un archivo debemos de hacer lo siguiente:

mando `2>archivo`

El *mando* puede ser cualquiera y con `2>` decimos que los errores se envíen al *archivo*.

Veamos un ejemplo

```
maisha@maisha-VirtualBox:~$ sort archivo
sort: open failed: archivo: No such file or directory
```

Figura 5: Uso del comando **sort** que envía un error porque no encuentro el archivo solicitado

El comando `sort` marca un error debido a que el archivo no se encuentra, este error lo podemos enviar a un archivo como se muestra en la figura 6.

```
maisha@maisha-VirtualBox:~$ sort archivo 2>error
maisha@maisha-VirtualBox:~$ ls
Desktop      error          Music          ordenados.txt  Templates     test.txt
Documents    examples.desktop nombres.txt    Pictures        test          text.txt
Downloads    mimes          numeros       Public         test1         Videos
```

Figura 6: Uso del redireccionamiento de error

Se ejecuta el mismo comando `sort` sobre el mismo archivo, pero el error que se genera se ha enviado a el archivo *error*, después se ejecuta el comando `ls` para ver que efectivamente se ha creado el archivo *error*.

Si no queremos hacerle caso a los errores que surjan de ejecutar un comando entonces enviamos los errores a un archivo específico `/dev/null` este es un archivo especial que su longitud siempre es cero. Cualquier cosa que se escriba en él desaparecerá. Veamos el ejemplo anterior utilizando el comando `sort`, pero en lugar de crear el archivo *error* enviaremos los errores al archivo `/dev/null`.

```
maisha@maisha-VirtualBox:~/practicassLinux$ sort archivo
sort: open failed: archivo: No such file or directory
maisha@maisha-VirtualBox:~/practicassLinux$ sort archivo 2>/dev/null
maisha@maisha-VirtualBox:~/practicassLinux$
```

Figura 7: Redireccionamiento del error al archivo `/dev/null`.

Observe la primera línea, el comando `sort` envía un mensaje de error por que no encuentra el archivo, este error podemos ignorarlo enviándolo al archivo `/dev/null` como se muestra en la tercera línea.

También hay forma de redireccionar la salida agregando al final del archivo el resultado de un comando, esto se logra utilizando `>>`. Por ejemplo
Si ejecutamos en una terminal

```
ls >> test
```

Se agrega al final del archivo *test* el listado de los archivos y directorios.

En la siguiente imagen se muestra el funcionamiento de la línea anterior utilizando el comando `more` para visualizar el contenido del archivo

.

```
maisha@maisha-VirtualBox: ~
maisha@maisha-VirtualBox:~$ ls >> test
maisha@maisha-VirtualBox:~$ more test
Este comando escribe en el archivo test
Para salir del modo de escritura se presiona al mismo tiempo CTR - d
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
test
test1
test.txt
text.txt
Videos
maisha@maisha-VirtualBox:~$
```

Figura 8: Visualización de un archivo con el mando **more**

4. PROCEDIMIENTO (DESCRIPCIÓN)

Debe realizar estos pasos con el mando **noclobber** encendido.

- 1.- Utilizando el redireccionamiento de salida tome las primeras 20 líneas del archivo `/etc/passwd` y cree el archivo `datos`.
- 2.- Agregue al final del archivo la información de los derechos del archivo recién creado.
- 3.- Cree otro archivo con la información ordenada de forma ascendente del archivo `datos` que se llame `datosOrdenados`, el 5to. dato de cada línea del archivo será considerado para el ordenamiento.
- 4.- Cree un archivo llamado `misgrupos` con las 30 últimas líneas del archivo `/etc/group`
- 5.- Ordene el archivo recién creado por el primer campo. El nuevo archivo se llamará `informacionOrdenada`.
- 6.- Agregue a `datosOrdenados` el archivo `informacionOrdenada`.
- 7.- Cambie los derechos de los archivos `datosOrdenados` e `informacionOrdenada` para que sean de solo escritura.
- 8.- Utilizando el mando `cat` visualice el contenido de los archivos `datosOrdenados`, `informacionOrdenada` y `misgrupos` y envíe los errores a un archivo que se llame `errores`.
- 9.- Realice el paso anterior pero ahora envíe a el archivo `/dev/null`, ¿qué pasará con la información?
- 10.- Realizar los pasos necesarios para almacenar el listado de todos los archivos y directorios con el nombre y la longitud en kb que se encuentran en su cuenta, el archivo se llamará `listaDocumentos`.
- 11.- Agregar al archivo `listaDocumentos` el número de líneas y los derechos del mismo archivo.

A) EQUIPO NECESARIO

Maquina con linux ubuntu 12

MATERIAL DE APOYO

Acceso a la práctica

B) DESARROLLO DE LA PRÁCTICA

C) CÁLCULOS (SI APLICA) Y REPORTE

5. RESULTADOS Y CONCLUSIONES

6. ANEXOS

7. REFERENCIAS