

# SISTEMAS OPERATIVOS

## LABORATORIO N°14


Redireccionamiento y  
Tuberías



**DOCENTE**  
Luis Díaz Bravo




**CURSO**  
Sistemas operativos

	Laboratorio 14 - Redireccionamiento y Tuberías	Nro. DD-106
		Página 1

## I. OBJETIVOS

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

## II. SEGURIDAD

	<p><b>Advertencia:</b> En este laboratorio está prohibida la manipulación de hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.</p>
---	--

## III. 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.

## IV. RECURSOS

- Computadora con Vmware Workstation.
- Máquina virtual con CentOS 7.

## V. METODOLOGÍA PARA EL DESARROLLO DE LA TAREA

- El laboratorio se desarrolla de forma individual.

## VI. 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 Díaz.

### NOTA

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

- Ingresa 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
```

```
[root@khailmo-srv ~]# sudo useradd -m -s /bin/bash james  
[root@khailmo-srv ~]# sudo passwd james  
Cambiando la contraseña del usuario james.
```

```
Nueva contraseña:  
Vuelva a escribir la nueva contraseña:  
passwd: todos los tokens de autenticación se actualizaron exitosamente.  
[root@khailmo-srv ~]#
```

- Inicie una sesión en una consola o terminal con la cuenta de james.

```
Nueva contraseña:  
Vuelva a escribir la nueva contraseña:  
passwd: todos los tokens de autenticación se actualizaron exitosamente.  
[root@khailmo-srv ~]#
```

- Crear los siguientes directorios dentro del directorio hogar de james.
  - ventas
  - informes
  - documentos
  - documentos/privados/

```
[james@khailmo-srv ~]$ mkdir ventas informes documentos  
[james@khailmo-srv ~]$ cd documentos  
[james@khailmo-srv documentos]$ mkdir privados  
[james@khailmo-srv documentos]$
```

- 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

```
[james@khailmo -srv ~]$ cat > carta.txt
Ejemplo
[james@khailmo -srv ~]$ cat > carta1.txt
Prueba
[james@khailmo -srv ~]$ cat > carta2.txt
Ejemplo
[james@khailmo -srv ~]$ cat > trabajo1.txt
Ejemplo
[james@khailmo -srv ~]$ cat > trabajo2.txt
Prueba
[james@khailmo -srv ~]$ cd documentos
[james@khailmo -srv documentos]$ cat > trabajo3.txt
Ejemplo
[james@khailmo -srv documentos]$ _
```

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

```
[james@khailmo -srv ~]$ ls
carta1.txt carta2.txt carta.txt documentos informes trabajo1.txt trabajo2.txt ventas
[james@khailmo -srv ~]$
```

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

```
[james@khailmo -srv ~]$ ls -lh | grep -v "^d" > lista.txt
```

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

```
[james@khailmo -srv ~]$ cat lista.txt
total 20K
-rw-rw-r--. 1 james james 7 nov 22 19:23 carta1.txt
-rw-rw-r--. 1 james james 8 nov 22 19:24 carta2.txt
-rw-rw-r--. 1 james james 8 nov 22 19:22 carta.txt
-rw-rw-r--. 1 james james 0 nov 22 19:51 lista.txt
-rw-rw-r--. 1 james james 8 nov 22 19:25 trabajo1.txt
-rw-rw-r--. 1 james james 7 nov 22 19:25 trabajo2.txt
[james@khailmo -srv ~]$
```

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

```
[james@khailmo-srv ~]$ ls trabaj* > lista.txt
[james@khailmo-srv ~]$ _
```

- Visualice el contenido del archivo:

```
[james@khailmo-srv ~]$ cat lista.txt
trabajo1.txt
trabajo2.txt
[james@khailmo-srv ~]$ _
```

- ¿Qué pasó con el contenido anterior?

Como se puede observar el contenido anterior ya no se encuentra en este archivo , solo aparece el la nueva lista en viada.

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

```
$ ls carta* >> lista.txt
[james@khailmo-srv ~]$ ls carta* >> lista.txt
[james@khailmo-srv ~]$
```

- ¿Qué pasó con el contenido anterior?

```
[james@khailmo-srv ~]$ cat lista.txt
trabajo1.txt
trabajo2.txt
carta1.txt
carta2.txt
carta.txt
[james@khailmo-srv ~]$ _
```

Como se puede observar , en esta ocasión el contenido anterior no se borro, el listado actual se agregó al anterior.

- 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
[james@khailmo-srv ~]$ w > salida1.txt
[james@khailmo-srv ~]$ cat salida1.txt
20:14:25 up 1:40, 1 user, load average: 0,00, 0,00, 0,00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
james     tty1    -              18:48    1.00s  0.19s  0.01s  w
[james@khailmo-srv ~]$
```

```
$ find /etc -name motd > salida2.txt
[james@khailmo-srv ~]$ find /etc -name motd > salida2.txt
find: '/etc/pki/rsyslog': Permiso denegado
find: '/etc/grub.d': Permiso denegado
find: '/etc/lvm/archive': Permiso denegado
find: '/etc/lvm/backup': Permiso denegado
find: '/etc/lvm/cache': Permiso denegado
find: '/etc/polkit-1/rules.d': Permiso denegado
find: '/etc/polkit-1/localauthority': Permiso denegado
find: '/etc/sssd': Permiso denegado
find: '/etc/dhcp': Permiso denegado
find: '/etc/firewalld': Permiso denegado
find: '/etc/audit': Permiso denegado
find: '/etc/sudoers.d': Permiso denegado
[james@khailmo-srv ~]$
```

- ¿Qué pasó en el primer comando?

En este caso, **w** devuelve una lista de todos los usuarios con sesión abierta y todos los procesos que están ejecutando.

- ¿Qué pasó en el segundo comando?

Al intentar aplicar el comando se me fue denegado.

### 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
```

```
[james@khailmo-srv ~]$ ls -l usuarios.txt
-rw-rw-r--. 1 james james 975 nov 22 20:35 usuarios.txt
[james@khailmo-srv ~]$ _
```

- 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?

```
[james@khailmo-srv ~]$ ls -l usuarios.txt
-rw-rw-r--. 1 james james 0 nov 22 20:38 usuarios.txt
[james@khailmo-srv ~]$
```

- 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 creo un archivo con 36 MB

```
[james@khailmo-srv ~]$ dd if=/dev/zero of=Archivo40MB.txt bs=6000 count=6000
6000+0 registros leídos
6000+0 registros escritos
36000000 bytes (36 MB, 34 MiB) copied, 0,0639402 s, 563 MB/s
```

#### Luego se procedió a realizar su búsqueda.

```
[james@khailmo-srv ~]$ find . -size +20000k -and -size -40000k
./Archivo40MB.txt
[james@khailmo-srv ~]$
```

- Ejecute el siguiente comando:

```
$ split <archivo copiado> -b 4MB
```

```
[james@khailmo-srv ~]$ split Archivo40MB.txt -b 4MB
[james@khailmo-srv ~]$
```

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

```
[james@khailmo-srv ~]$ ls xa*
xaa xab xac xad xae xaf xag xah xai
```

- ¿Qué sucede si ejecuta el siguiente comando?

```
$ split <archivo copiado> -b 4MB part_
```

- ¿Qué sucede si ejecuta el siguiente comando?

```
$ cat part_* > arcuno
```

```
[james@khailmo-srv home]$ cat part_* > arcuno  
-bash: arcuno: Permiso denegado  
[james@khailmo-srv home]$
```

- 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 esto 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
```

```
khailmo-srv login: root  
Password:  
Last login: Fri Nov 22 20:25:14 on tty1  
[root@gsalas-srv ~]# sudo useradd -m -s /bin/bash fgarcia  
[root@gsalas-srv ~]# sudo passwd fgarcia  
Cambiando la contraseña del usuario fgarcia.  
Nueva contraseña:  
Vuelva a escribir la nueva contraseña:  
passwd: todos los tokens de autenticación se actualizaron exitosamente.  
[root@khailmo-srv ~]#
```

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

```
khailmo-srv login: fgarcia  
Password:  
[fgarcia@gsalas-srv ~]$ who  
fgarcia  tty1          2019-11-22 22:06  
james   tty2          2019-11-22 22:02  
[fgarcia@khailmo-srv ~]$
```

- 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
```



```
'mensaje" [New] 1L, 88C written
[james@khailmo-srv ~]$ write fgarcia < mensaje
[james@khailmo-srv ~]$
```

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

```
[fgarcia@khailmo-srv ~]$
Mensaje de james@khailmo-srv el tty2 a las 22:22 ...
Este es un archivo de prueba de c\303\263mo se utiliza el car\303\241cter de rendici\303\263n de ent
rada.
EOF
```

- 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**

```
[james@khailmo-srv ~]$ ps -aux | tee 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?

```
[james@khailmo-srv ~]$ who | tee --append procesos.txt
fgarcia  tty1          2019-11-22 22:06
james    tty2          2019-11-22 22:02
```

## 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

[root@khailmo-srv home]# cd ..
[root@khailmo-srv /]# cd /home
[root@khailmo-srv home]# sudo mkdir prueba

[root@khailmo-srv home]# chmod 123 prueba
[root@khailmo-srv home]# _
```

**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
```

```
[james@khailmo-srv ~]$ ls anual
ls: no se puede acceder a 'anual': No existe el fichero o el directorio
[james@khailmo-srv ~]$
```

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

Si , por que no existe el fiche o directorio.

```
$ find /home -name hola
[james@khailmo-srv ~]$ find /home -name hola
find: '/home/khailmo': Permiso denegado
find: '/home/bsalas': Permiso denegado
find: '/home/jperez': Permiso denegado
find: '/home/fgarcia': Permiso denegado
find: '/home/prueba': Permiso denegado
```

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

Debido que no cuento con los permisos correspondientes.

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

```
[james@khailmo-srv ~]$ ls anual 2> errores.txt
[james@khailmo-srv ~]$ cat errores.txt
```

- ¿Qué información es la que se muestra?

```
[james@khailmo-srv ~]$ cat errores.txt
ls: no se puede acceder a 'anual': No existe el fichero o el directorio
[james@khailmo-srv ~]$
```

- Verifique el contenido de los archivos:

```
$ cat error-ls
$ cat error-find
```

```
[james@khailmo-srv ~]$ cat error-ls
cat: error-ls: No existe el fichero o el directorio
[james@khailmo-srv ~]$ cat error-find
cat: error-find: No existe el fichero o el directorio
[james@khailmo-srv ~]$ _
```

- ¿Qué contienen estos archivos?

Mensajes de error.

- Redireccione ahora también la salida del comando

```
$ find /home -name mensaje 1>resul2.txt 2>error2.txt
[james@khailmo-srv ~]$ cat error2.txt
find: '/home/khailmo': Permiso denegado
find: '/home/bsalas': Permiso denegado
find: '/home/jperez': Permiso denegado
find: '/home/fgarcia': Permiso denegado
find: '/home/prueba': Permiso denegado
[james@gsalas-srv ~]$
```

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

```
[james@khailmo-srv ~]$ cat error2.txt
find: '/home/khailmo': Permiso denegado
find: '/home/bsalas': Permiso denegado
find: '/home/jperez': Permiso denegado
find: '/home/fgarcia': Permiso denegado
find: '/home/prueba': Permiso denegado
[james@gsalas-srv ~]$
```

- 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
[james@khailmo-srv ~]$ find /home -name mensaje 1>resul3.txt 2>&1
[james@khailmo-srv ~]$ _
```

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

```
[james@khailmo-srv ~]$ cat resul3.txt
find: '/home/khailmo': Permiso denegado
find: '/home/bsalas': Permiso denegado
find: '/home/jperez': Permiso denegado
/home/james/mensaje
find: '/home/fgarcia': Permiso denegado
find: '/home/prueba': Permiso denegado
[james@gsalas-srv ~]$
```

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

```
james@khailmo-srv ~]$ ls | tee 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**.

```
$ wc _____
```

```
1 archivo1.txt
2 Archivo20MB.txt
3 Archivo21MB.txt
4 Archivo25MB.txt
5 Archivo30MB.txt
6 Archivo40MB.txt
7 archivo.txt
8 carta1.txt
9 carta2.txt
10 carta.txt
11 documentos
12 error2.txt
13 errores.txt
14 informes
15 lista2.txt
16 lista.txt
17 mensaje
18 procesos.txt
19 prueba.txt
20 resul2.txt
21 resul3.txt
22 salida1.txt
23 salida2.txt
24 trabajo1.txt
25 trabajo2.txt
26 usuarios.txt
27 ventas
28 xaa
29 xab
30 xac
31 xad
32 xae
33 xaf
34 xag
35 xah
36 xai
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
adjtime
aliases
alternatives
anacrontab
audit
authselect
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
centos-release
centos-release-upstream
chkconfig.d
cifs-utils
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
crontab
cron.weekly
crypto-policies
crypttab
csh.cshrc
csh.login
dbus-1
default
depmod.d
dhcp
DIR_COLORS
DIR_COLORS.256color
DIR_COLORS.lightbgcolor
dnf
dracut.conf
dracut.conf.d
environment
--Más--
```

- 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.

```
[james@khailmo-srv ~]$ who | sort
fgarcia  tty1      2019-11-22 22:06
james    tty2      2019-11-22 22:02
root     tty3      2019-11-22 22:36
[james@khailmo-srv ~]$ _
```

- 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):

```
[root@khailmo-srv home]# date +%y%m%d -s 20191022
191022
[root@gsalas-srv home]# date
mar oct 22 00:00:04 -05 2019
[root@khailmo-srv home]# _
```

- 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
```

```
[james@khailmo-srv ~]$ ls -l | grep ago
```

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

```
[root@khailmo-srv home]# hwclock --hctosys
[root@khailmo-srv home]# date
vie nov 22 23:37:43 -05 2019
[root@khailmo-srv home]# _
```

- 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 esto?

- 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?

- 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.

## VII. CONCLUSIONES

- Las redirecciones consisten en trasladar información de un tipo de fichero a otro, por ejemplo, de error estándar a la salida estándar o de la salida estándar a la entrada estándar.
- las tuberías funcionan de manera muy similar, en este caso, nos permiten ingresar un elemento de entrada el cual podrá ser conectado con varios comandos o procesos de manera secuencial antes de obtenerse la salida final.
- Se pueden utilizar de manera conjunta las redirecciones y las tuberías obteniendo consultas más específicas y complejas

