

Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
Laboratorio # 6 - Sistemas Operativos I

Carlos D. Gonzalez F. 20-70-5162

Objetivos.

- Crear enlaces simbólicos de un archivo o directorio.
- Comandos: su, ln, chmod, chown, umask,
- Uso de pipe

1. Comando su

Switch user. Permite iniciar una sesión con otro usuario.

Ejemplo:
su root

2. Comando ln

Hard Link

El comando ln permite que un archivo en el disco sea accedido con más de un nombre de archivo a través de un enlace duro. El comando crea un nuevo archivo y lo conecta con el original. Permite que dos usuarios compartan el mismo archivo o accederlo cuando se encuentra dentro de una cadena larga de subdirectorios.

Sintaxis:

ln ruta_archivo nombre_acceso

Ejemplo:

ln /home/trabajos/tareas/formula.txt formulatemp

Procedimiento:

- Cree el archivo formula.txt en dir1 el cual está en su directorio regular de trabajo (No. Grupo). Si no existen los directorios, créelos.
- El archivo formula.txt contiene las fórmulas de área de un triángulo, rectángulo y círculo.
- Desde el directorio de trabajo (~) cree el enlace
- Añada una línea al archivo formula.txt usando el enlace.
- Verifique el contenido de ambos archivos – formula.txt y formulatemp.txt
- Borre el enlace
- Escriba cada uno de los comandos para realizar estas tareas.

Nota: formulatemp es un enlace al archivo formula.txt el cual se encuentra dentro de toda la cadena de directorios /home/trabajos/tareas/. Sólo existe una copia del archivo, pero puede ser accedido rápidamente por el enlace. Cuando se trate de borrar el archivo, Linux no lo permitirá mientras exista el enlace.

Symbolic Link

Permite crear un enlace a un directorio. También es requerido cuando se desee enlazar un archivo en otra partición de disco o en la red. Para crear un enlace simbólico se debe agregar el parámetro -s al comando ln.

ln -s /dev/fd0 A:

ln -s /etc/httpd/conf web

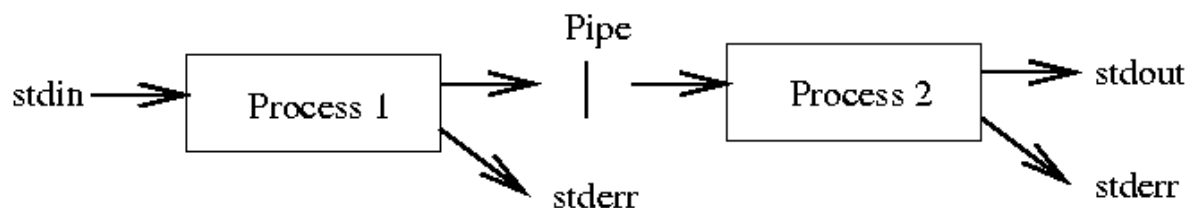
El primer ejemplo permite acceder al floppy como A y el segundo crea un enlace al directorio web en lugar de escribir toda la ruta `/etc/httpd/conf`.

Ejercicio:

- Ubíquese en el directorio de trabajo de su usuario
- Haga un enlace simbólico a su directorio `dir1` localizado dentro de su directorio de trabajo (no. Grupo). Nómbrelo `midir`
- Entre al enlace recién creado
- Liste el contenido del directorio
- Haga un enlace a su unidad usb (ver `/media` para confirmar que la unidad está montada)
- Borre los enlaces
- Escriba cada uno de los comandos para realizar estas tareas.

3. Uso de filtro o pipe (|)

El pipe permite enlazar la salida de varios comandos juntos. Permite que procesos separados se comuniquen sin que explícitamente estén diseñados para trabajar juntos y que funcionen en forma compleja.



El símbolo “|” especifica que toda la salida (stdout) será entubada como entrada para el siguiente comando.

Ejemplo 1:

```
cd /lib
ls -l | more
ls -l | grep read
```

Ejemplo 2:

```
ps aux | grep sort
```

La salida nos permite ver específicamente si la aplicación `sort` está en ejecución. El comando `ps` brinda la lista de los procesos corriendo, y con `grep` se reduce la salida a unos cuantos procesos.

Ejemplo 3:

```
ls / | grep x
```

Bash y otros shells corren ambos comandos, conectando la salida del primero a la entrada del segundo; `ls` produce la lista de archivos en el directorio actual, mientras que `grep` toma dicha lista e imprime sólo aquellas líneas que contengan la letra `x`.

4. Comando chmod

Permite cambiar los permisos a archivos.

Sintaxis:

```
chmod [option] archivo
```

```
u  permisos del usuario (dueño)
g  permisos del grupo
o  permisos de otros (público)
a  todos los usuarios, permisos de otros y del grupo
=  asigna un permiso absolutamente
```

- + agrega un permiso
- retira un permiso

Ejemplos:

```
chmod u+x test1
chmod 700 test2
```

Con la primera línea se ha dado permiso de ejecución al dueño del archivo test1 y con la segunda se ha hecho lo mismo, usando representación octal, para el archivo test2 al dueño.

5. Comando umask:

Establece los permisos por defecto. Debe darse en representación octal:

| | Owner | Group | Public (others) |
|--------|-------|-------|-----------------|
| | r w x | r w x | r w x |
| R ---- | 400 | 40 | 4 |
| - | | | |
| W -- | 200 | 20 | 2 |
| --- | | | |
| X -- | 100 | 10 | 1 |
| --- | | | |

Ejemplo:

```
umask 644
```

```
400
200
40
4
---
644
```

644 es rw-r--r--

El dueño puede leer y escribir el archivo o directorio, los demás sólo pueden leerlo.

Numeración octal para algunos permisos comunes:

755 es rwxr-xr-x

El dueño puede leer, escribir y ejecutar el archivo, los demás sólo pueden leer o ejecutarlo. Para un directorio, esto es equivalente a 644.

711 es rwx--x--x

El dueño puede leer, escribir y ejecutar el archivo, los demás sólo pueden ejecutarlo.

444 es r--r--r--

Este permiso significa solo lectura para todos.

6. Comando chown

Cambia el dueño y el grupo a un archivo.

```
chown [OPTION]... OWNER[:[GROUP]] FILE...
chown [OPTION]... :GROUP FILE...
```

Opciones:

`-R, --recursivo: opera en archivos y directorios recursivamente.`

Ejemplo:

```
chown estudiantel test1
```

Ahora el dueño del archivo test1 es estudiantel. Todos los permisos del archivo pasan al nuevo dueño - usuario estudiantel.

7. Comando echo:

Despliega una línea de texto.

Ejemplo:

```
#echo hello
#echo $HOME
```

8. which:

Muestra la ruta completa de los comandos del shell.

Ejemplo:

```
# which emacs
/usr/bin/emacs
```

9. whereis:

Busca en una serie de lugares estándares de Linux. Muestra el binario, el fuente, y las páginas de ayuda del archivo o comando.

Ejemplo:

```
# whereis ifconfig
```

Variables de Ambiente

env: lista las variables de ambiente.

Ejemplo:

```
#env
.
.
.
USERNAME=aris
MAIL=/var/spool/aris
PATH=/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/home/aris/bin
DESKTOP_SESSION=gnome
PWD=/home/aris
LANG=en_US.utf8
.
.
.
```

\$PATH: Environment variable that stores the search path for executables.

Ejemplo:

```
echo $PATH
/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/home/aris/bin
```

\$HOME:

export: Establece una variable de ambiente en el bash shell.

Ejemplo:

```
export PATH=$PATH:/sbin
```

Ejercicios:

1. Muestre en pantalla en formato completo el contenido del directorio etc, pantalla por pantalla.
2. Busque los archivos con caracteres host en el directorio etc y que le detalle la cantidad de líneas de los mismos.
3. Visualice el contenido del directorio /usr/bin pantalla por pantalla.
4. Cree un archivo en vi con 5 líneas de texto (use echo en cada línea). Luego cambie los permisos de lectura, escritura y ejecución del mismo. Ejecútelo.
5. Cámbiese a otro usuario. Pruebe si puede ejecutar el archivo. Realice las pruebas necesarias para verificar los permisos asignados al archivo.
6. Cambie el dueño del archivo (debe hacerlo con el usuario root). Realice todas las pruebas de verificación nuevamente.
7. Anote todos los pasos y comandos utilizados.

Desarrollo

Punto 2 Ejercicios:

```
cd ~
ln -s dir1 midir
cd medir
ls
ls /media
ln -s /media/<CARLOS> CARLOS
rm midir
rm CARLOS
```

Variables de Ambiente Ejercicios:

```
1. less -R /etc/*
2. grep -r "host" /etc/ | wc -l
3. less -R /usr/bin/*
4. vi archivo.txt
   chmod 700 archivo.txt
   ./archivo.txt
5. su User2
   ./archivo.txt
6. sudo chown root archivo.txt
7. less -R /etc/*
   grep -r "host" /etc/ | wc -l
   less -R /usr/bin/*
   vi archivo.txt
   chmod 700 archivo.txt
   ./archivo.txt
   su User2
   ./archivo.txt
   sudo chown root archivo.txt
```