

Universidad Tecnológica de Panamá
Sistemas Operativos I
Experiencia Práctica en Laboratorio No. 3
Manejo de Archivos

Prof. Aris Castillo de Valencia

Estudiante: Manuel Villanueva

Cédula: E-8-172111

Objetivo:

- Probar y distinguir distintos comandos para trabajar con archivos en línea de texto, incluyendo: moverse entre carpetas, buscar información dentro de archivos de texto, agregar texto, desplegar en pantalla el contenido de archivos y realizar copias de seguridad.

Metas:

Con esta experiencia práctica se espera que el estudiante sea capaz de realizar tareas sencillas de administración del sistema operativo Linux/GNU a través de comandos para manipular archivos.

Contenidos:

- Comandos de manejo de archivos: cd, cat, less, more, head, tail, find, grep, ls, file, cpio.

Metodología:

Se basa en métodos intuitivos, de experimentación y demostración en que se acerca al estudiante a situaciones reales de la práctica profesional de manera que resuelva las situaciones presentadas.

Evaluación:

- Se dará 50 puntos por el desarrollo de la práctica en el aula.
- Se dará 50 puntos por la entrega del informe escrito debidamente completado y por su nivel técnico.

Recursos:

- Hardware: computadora, conexión a Internet.

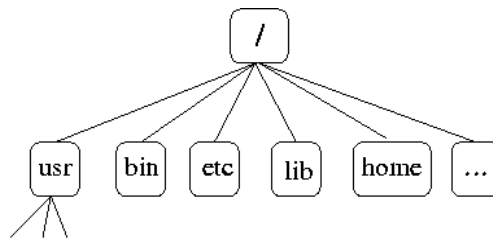
- Software: Sistema operativo Linux/GNU.

Procedimiento:

Lea cuidadosamente la guía; pruebe cada uno de los comandos listados prestando especial atención a los resultados obtenidos y a las variantes que le ofrecen las opciones de los comandos. Ponga en práctica los comandos aprendidos haciendo los ejercicios sugeridos. Llene la autoevaluación y retroalimentación y súbala a la plataforma Moodle.

¿Cómo me muevo entre carpetas?

El comando `cd` (change directory) le permite cambiar de directorio. Son muchas las opciones que puede utilizar dependiendo de lo que desea realizar. Trabajaremos con el árbol de directorio de Linux, que es clave para su comprensión de este sistema operativo.



Recuerde que Linux es un sistema operativo **multiusuario**. Para cada usuario creado, el sistema le crea una carpeta de trabajo, comúnmente llamada **“home,”** las cuales estarán ubicadas dentro del directorio “home.” Tenga presente que, generalmente, cuando se habla de ir a “home” no se trata de la carpeta “home” dentro del directorio raíz “/” sino de *la carpeta del usuario actualmente en sesión*. Por ejemplo, si usted es el usuario X, habrá una carpeta llamada X, que contendrá todos sus archivos y directorios. Si existe otro usuario Y en el sistema, habrá otra carpeta Y, que contendrá todos los archivos y directorios del usuario Y. Así sucesivamente ocurrirá para cada usuario existente. Eso es lo que comúnmente se le llama “home.”

Sintaxis: **`cd nombre_directorio`**

Resultado: se pasa a la carpeta nombre_directorio

Algunos ejemplos:

Antes de realizar los ejemplos, cree una carpeta llamada laboratorio2. Dentro de ésta cree dos subcarpetas – tips y practicas. Dentro de tips cree una carpeta llamada cocina.

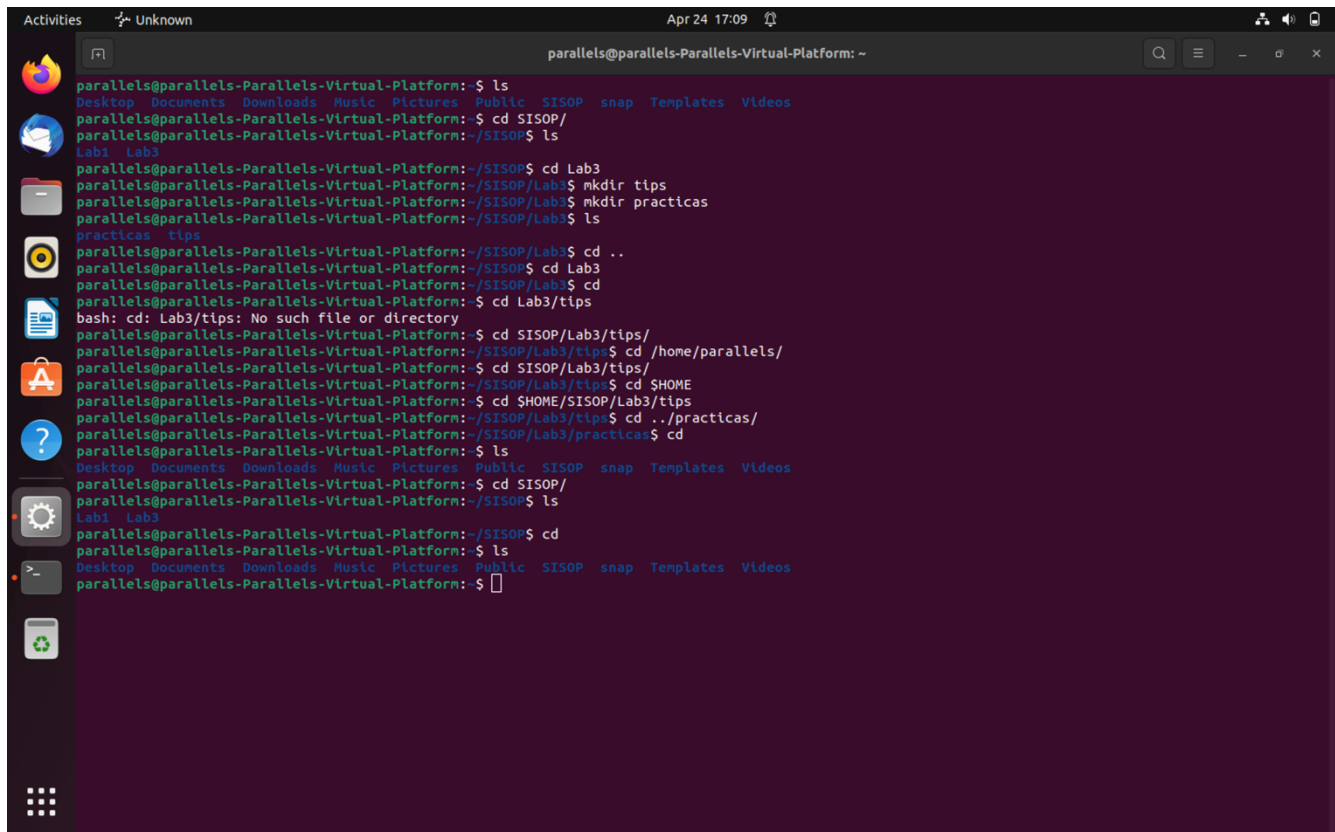
Comando ejecutado	Resultado
cd laboratorio2	Se pasa del directorio home del usuario actualmente en sesión a la carpeta laboratorio2. Note que Linux le coloca el nombre del directorio donde se encuentra actualmente.
cd	Permite regresar al directorio home del usuario actualmente en sesión
cd laboratorio2/tips	Se pasa en un sólo paso del directorio home del usuario actualmente en sesión a la carpeta tips que se encuentra dentro de la carpeta laboratorio2
cd ..	Se mueve hacia atrás, es decir, al directorio padre del directorio actual. En este caso como la carpeta tips se encontraba dentro de la carpeta laboratorio2, Linux lo deja en ésta última.
cd /home/aris	Se mueve al directorio home de mi usuario. En su caso, reemplace aris por el nombre de su usuario. Recuerde que para moverse al directorio home de su usuario, no es necesario colocar toda la ruta, sino solo ejecutar cd. Esto sucede pues el directorio por default es el del usuario en sesión.
cd \$HOME	Hace lo mismo que la secuencia anterior. Se mueve al directorio home del usuario en sesión. En este caso \$HOME es una variable de ambiente.
cd \$HOME/laboratorio2/tips	Lo lleva al directorio tips. Cuál sería la forma utilizando la ruta completa?
cd ../practicas	Lo lleva al directorio practicas que está dentro de laboratorio2. En este caso ../ está repitiendo la ruta del directorio padre del directorio tips donde nos encontrábamos antes de ejecutar el comando. Cuál sería la otra alternativa?

- Pase al directorio raíz. Luego liste su contenido. Muévase al directorio /home. Liste su contenido. Regrese al directorio home de su usuario. Liste la secuencia de comandos ejecutados de forma completa.

Nota especial:

Es preferible que los nombres de archivos y directorios en Linux no tengan caracteres especiales ni espacios en blanco. Sin embargo, si se da el caso, para pasarse a un directorio que contenga un espacio en blanco en el nombre habría que omitir dicho espacio utilizando la barra inversa. Así, si quiero entrar a un directorio llamado “aris castillo,” el comando cd debería ejecutarse de la siguiente manera:

```
cd aris\ castillo
```



```
parallels@parallels-Parallels-Virtual-Platform: ~  
$ ls  
Desktop Documents Downloads Music Pictures Public SISOP snap Templates Videos  
parallels@parallels-Parallels-Virtual-Platform: ~$ cd SISOP/  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ ls  
Lab1 Lab3  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ cd Lab3  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3$ mkdir tips  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3$ mkdir practicas  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3$ ls  
practicas tips  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cd ..  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ cd Lab3  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cd  
parallels@parallels-Parallels-Virtual-Platform: $ cd Lab3/tips  
bash: cd: Lab3/tips: No such file or directory  
parallels@parallels-Parallels-Virtual-Platform: $ cd SISOP/Lab3/tips/  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3/tips$ cd /home/parallels/  
parallels@parallels-Parallels-Virtual-Platform: $ cd SISOP/Lab3/tips/  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3/tips$ cd $HOME  
parallels@parallels-Parallels-Virtual-Platform: $ cd $HOME/SISOP/Lab3/tips  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3/tips$ cd ../practicas/  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3/practicas$ cd  
parallels@parallels-Parallels-Virtual-Platform: $ ls  
Desktop Documents Downloads Music Pictures Public SISOP snap Templates Videos  
parallels@parallels-Parallels-Virtual-Platform: $ cd SISOP/  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ ls  
Lab1 Lab3  
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ cd  
parallels@parallels-Parallels-Virtual-Platform: $ ls  
Desktop Documents Downloads Music Pictures Public SISOP snap Templates Videos  
parallels@parallels-Parallels-Virtual-Platform: $
```

¿Cómo puedo ver el contenido de un archivo en pantalla?

Para ver el contenido de un archivo se puede utilizar el comando **CAT (Concatenar)**, así **cat file1**. El resultado es que muestra en pantalla el contenido de file1.

También puede hacer esto con el comando **less**, así: **less file1**. La ventaja del comando less es que va mostrando pantalla por pantalla. El usuario presiona la barra espaciadora para ir avanzando y **q** para terminar. Este comando es útil cuando el archivo es muy largo.

Si lo que se desea es ver sólo el principio del archivo o el final, se pueden utilizar los comandos **head** o **tail**, así: **head x file1**; donde x es el número de líneas que desea ver, sino se especifica el comando le mostrará las 10 primeras líneas del archivo. Con el comando **tail**, sería: **tail x file1**. En este caso, x serían las últimas líneas del archivo. Si no se especifica, se mostrarán las 10 últimas líneas.

¿Se puede ver el contenido de varios archivos a la vez?

Se utiliza el comando cat colocando los nombres de los archivos uno después del otro, así: **cat file1 file2 file3**.

Resultado: Muestra en pantalla el contenido de cada uno de los archivos listados uno después del otro.

Resultado: Concatena archivos o entrada estándar a la salida estándar. Los resultados son diversos dependiendo de la opción utilizada. Veamos

¿Cómo creo un archivo y le agrego datos?

Para esto se puede utilizar el comando cat con direccionamiento desde la entrada estándar. Sintaxis: **cat > file1**.

Resultado: Crea el archivo file1 y le permite agregar texto por el teclado. Para culminar la escritura pulse Ctrl+x. La entrada estándar se refiere a.

Si ya he cerrado el archivo, ¿cómo le agrego más datos?

Para esto también se puede utilizar el comando cat, utilizando doble direccionamiento. Sintaxis: **cat >> file1**.

Resultado: Permite agregar líneas de texto a un archivo existente sin modificar su contenido.

¿Cómo buscamos archivos cuando sólo tenemos cierta información de los mismos?

El comando **find** (encontrar) es sumamente útil para buscar de distintas formas archivos cuando tenemos sólo algunas pistas para proporcionar al sistema operativo.

Sintaxis: **find opción**

Resultado: Busca archivos dentro de una jerarquía de directorios.

Ejemplos:

Ejemplo	Resultado
find / -name windows	Busca el archivo "windows" desde el directorio raíz "/" La opción name hace que el nombre sea sensitivo a las mayúsculas. Si no está seguro, puede usar la opción iname.
find /home -user aris	Encuentra cualquier archivo debajo del directorio home que pertenezca al usuario (opción -user) aris.

<code>find /usr -name *lab*</code>	Encuentra cualquier archivo debajo del directorio usr que tenga lab como parte del nombre (opción -name).
<code>find /home mtime +120</code>	Encuentra cualquier archivo debajo del directorio home que tenga 120 o más días de modificación (opción -mtime).

** Más ejemplos en:

http://linux.about.com/od/commands/a/blcmdl1_findx.htm

- Haga sus propias pruebas para aplicar el comando. Escriba 5 ejemplos completos que funcionen de acuerdo con la configuración del equipo. En cada caso explique qué deseaba lograr y cada parámetro utilizado.

find -name SISOP Deseaba encontrar la carpeta SISOP donde almaceno mis laboratorios de la clase.

find -name tips Deseaba encontrar la carpeta Tips dentro de mi directorio SISOP

find /home -user parallels Ver que archivos están debajo de home pertenecientes a mi usuario de máquina virtual.

find /home mtime +30 Ver archivos debajo de home con más de 30 días de modificación

find /usr -name *tips* Ver archivos debajo de usr que tengan tips en el nombre

```

parallels@parallels-Parallels-Virtual-Platform: ~/SISOP
$ find -name SISOP
./SISOP
parallels@parallels-Parallels-Virtual-Platform: $ cd SISOP/
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ find -name tips
./Lab3/tips
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP$ find /home -user parallels
/home/parallels
/home/parallels/SISOP
/home/parallels/SISOP/Lab3
/home/parallels/SISOP/Lab3/tips
/home/parallels/SISOP/Lab3/practicass
/home/parallels/SISOP/Lab1
/home/parallels/SISOP/Lab1/files.tar
/home/parallels/SISOP/Lab1/file-2.tar.bz2
/home/parallels/SISOP/Lab1/file3
/home/parallels/SISOP/Lab1/file-3.tar.bz2
/home/parallels/SISOP/Lab1/file1
/home/parallels/SISOP/Lab1/file4
/home/parallels/SISOP/Lab1/file2
/home/parallels/SISOP/Lab1/file.tar.gz
/home/parallels/.config
/home/parallels/.config/evolution
/home/parallels/.config/evolution/sources
/home/parallels/.config/evolution/sources/system-calendar.source
/home/parallels/.config/evolution/sources/system-proxy.source
/home/parallels/.config/evolution/sources/birthdays.source
/home/parallels/.config/monitors.xml
/home/parallels/.config/.gsd-keyboard.settings.ported
/home/parallels/.config/update-notifier
/home/parallels/.config/user-dirs.dirs
/home/parallels/.config/gnome-session
/home/parallels/.config/gnome-session/saved-session
/home/parallels/.config/nautilus
/home/parallels/.config/dconf
/home/parallels/.config/dconf/user
/home/parallels/.config/ibus
/home/parallels/.config/ibus/bus
/home/parallels/.config/ibus/bus/7680e13d7f0c23458e1541f2f62c4d40-unix-0
/home/parallels/.config/ibus/bus/7680e13d7f0c23458e1541f2f62c4d40-unix-wayland-0
/home/parallels/.config/gtk-3.0
/home/parallels/.config/gtk-3.0/bookmarks
/home/parallels/.config/user-dirs.locale
/home/parallels/.config/gnome-initial-setup-done
/home/parallels/.config/goa-1.0
/home/parallels/.config/pulse
/home/parallels/.config/pulse/7680e13d7f0c23458e1541f2f62c4d40-default-sink

```

Experiencia Práctica en Laboratorio No. 3

```
Activities Unknown Apr 24 17:30
parallels@parallels-Parallels-Virtual-Platform: ~/SISOP
parallels@parallels-Parallels-Virtual-Platform:~/SISOP$ find /home mtime +30
/home
/home/parallels
/home/parallels/SISOP
/home/parallels/SISOP/Lab3
/home/parallels/SISOP/Lab3/tips
/home/parallels/SISOP/Lab3/practicas
/home/parallels/SISOP/Lab1
/home/parallels/SISOP/Lab1/files.tar
/home/parallels/SISOP/Lab1/file-2.tar.bz2
/home/parallels/SISOP/Lab1/File3
/home/parallels/SISOP/Lab1/file-3.tar.bz2
/home/parallels/SISOP/Lab1/File1
/home/parallels/SISOP/Lab1/File4
/home/parallels/SISOP/Lab1/File2
/home/parallels/SISOP/Lab1/file.tar.gz
/home/parallels/.config
/home/parallels/.config/evolution
/home/parallels/.config/evolution/sources
/home/parallels/.config/evolution/sources/system-calendar.source
/home/parallels/.config/evolution/sources/system-proxy.source
/home/parallels/.config/evolution/sources/birthdays.source
/home/parallels/.config/monitors.xml
/home/parallels/.config/.gsd-keyboard.settings-ported
/home/parallels/.config/update-notifier
/home/parallels/.config/user-dirs.dirs
/home/parallels/.config/gnome-session
/home/parallels/.config/gnome-session/saved-session
/home/parallels/.config/nautilus
/home/parallels/.config/dconf
/home/parallels/.config/dconf/user
/home/parallels/.config/ibus
/home/parallels/.config/ibus/bus
/home/parallels/.config/ibus/bus/7680e13d7f0c23458e1541f2f62c4d40-unix-0
/home/parallels/.config/ibus/bus/7680e13d7f0c23458e1541f2f62c4d40-unix-wayland-0
/home/parallels/.config/gtk-3.0
/home/parallels/.config/gtk-3.0/bookmarks
/home/parallels/.config/user-dirs.locale
/home/parallels/.config/gnome-initial-setup-done
/home/parallels/.config/goa-1.0
/home/parallels/.config/pulse
/home/parallels/.config/pulse/7680e13d7f0c23458e1541f2f62c4d40-default-sink
/home/parallels/.config/pulse/7680e13d7f0c23458e1541f2f62c4d40-default-source
/home/parallels/.config/pulse/7680e13d7f0c23458e1541f2f62c4d40-card-database.tdb
/home/parallels/.config/pulse/cookie
/home/parallels/.config/pulse/7680e13d7f0c23458e1541f2f62c4d40-stream-volumes.tdb
```

```
Activities Terminal Apr 24 17:32
parallels@parallels-Parallels-Virtual-Platform: ~
parallels@parallels-Parallels-Virtual-Platform:~/SISOP$ find /usr -name *Lab3*
parallels@parallels-Parallels-Virtual-Platform:~/SISOP$ cd
parallels@parallels-Parallels-Virtual-Platform:~$ find /usr -name *Lab3*
parallels@parallels-Parallels-Virtual-Platform:~$ find /usr -name *tips*
/usr/share/help/C/gnome-help/tips.page
/usr/share/help/C/gnome-help/figures/tile-tips.svg
/usr/share/help/C/gnome-help/net-security-tips.page
/usr/share/help/C/gnome-help/tips-specialchars.page
/usr/share/locale-langpack/en_CA/LC_MESSAGES/gimp20-tips.mo
/usr/share/locale-langpack/en_AU/LC_MESSAGES/gimp20-tips.mo
/usr/share/locale-langpack/en_GB/LC_MESSAGES/gimp20-tips.mo
parallels@parallels-Parallels-Virtual-Platform:~$
```

¿Cómo busco, no archivos, sino información dentro de los archivos?

Para esto también se utiliza el comando **less**. En este caso, se aplica primero el comando less y luego de estar en pantalla, se presiona / (slash) y la palabra a buscar. Ejemplo: less practica. Con el archivo abierto, presione /comando. Presione n para continuar.

Nota: Puede limpiar la pantalla con el comando **clear** (limpiar).

También puede buscar dentro de un archivo por palabras específicas o patrones usando el comando **grep**.
Sintaxis: **grep option pattern file1**.

Resultado: Imprime en pantalla las líneas de file1 que cumplan con las opciones o patrón indicado.

Ejemplo: grep comando practica. En este caso el resultado es todas y cada una de las líneas del archivo practica en que aparezca la palabra comando.

El comando grep es **sensitivo a las mayúsculas**, así que para ignorar la mayúscula o minúscula, se debe usar la opción -i antes de la palabra a buscar, así:

```
grep -i comando practica
```

Si deseamos buscar una frase, en lugar de una sola palabra, enciérrela en apóstrofe simple, así:

```
grep -i 'spanning tree protocol' practica
```

Otras opciones del comando son:

v	para mostrar todas las líneas que no coinciden
n	coloca antes de cada línea el número de línea
c	presenta sólo el número total de líneas

¿Cómo determino el tipo de archivo?

A través de un **ls (list)** se podría determinar el tipo de archivo viendo la extensión de los mismos. Sin embargo, hay casos en que no hay una extensión, entonces se puede utilizar el comando **file**.

Así:

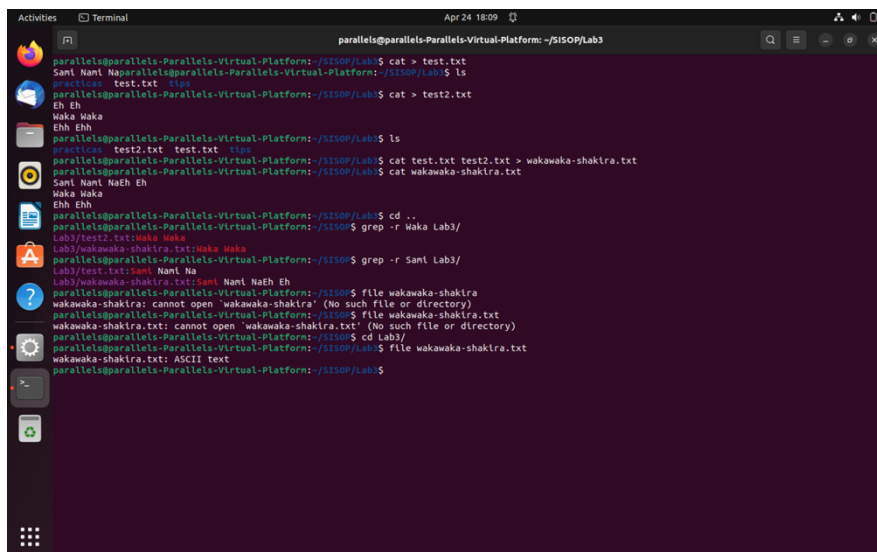
file nombre-archivo. Donde **nombre-archivo** es el archivo al cual le desea aplicar el comando para conocer su tipo.

El **formato de un archivo** depende de los programas que lo usan. Puesto que los tipos de archivos no son determinados por el sistema de archivos, el núcleo no puede decir cuál es el tipo, pues no lo conoce.

El **comando file** no atiende el nombre de los archivos, ya que variedad de convenciones de nombres. Los archivos con sufijo **.c** son programas fuentes en C, pero nada impide que su contenido sea arbitrario. Lo que hace el comando **file** es leer unos cuantos bytes al principio del archivo y busca los indicios que indiquen el tipo del archivo [1].

Por ejemplo, si usted ha creado un archivo con extensión **txt** probablemente el resultado de **file** le indicará que se trata de un archivo UTF-8 Unicode Text. Esto significa que se trata de una codificación de caracteres de 8 bits del set de caracteres Unicode, compatible con ASCII. Este sistema de codificación es el más generalizado en sistemas operativos, lenguajes de programación, APIs y aplicaciones de software [2].

3. Haga la prueba con algunos archivos. Cree archivos con el comando **cat**. Revise con el comando **file**. Describa las líneas completas ejecutadas. ¿Qué resultados obtuvo? Busque el significado de los tipos de archivos de los resultados obtenidos.



```
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cat > test.txt
Sant Nant
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ ls
parallel test.txt
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cat > test2.txt
EH Eh
Waka Waka
Ehh Ehh
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ ls
parallel test2.txt test.txt
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cat test.txt test2.txt > wakawaka-shakira.txt
Sant Nant NaTh Eh
Waka Waka
Ehh Ehh
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ cd ..
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP$ grep -r Waka Lab3/
Lab3/test2.txt:Waka Waka
Lab3/wakawaka-shakira.txt:waka waka
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP$ grep -r Sant Lab3/
Lab3/test.txt:Sant Nant Na
Lab3/wakawaka-shakira.txt:Sant Nant NaTh Eh
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP$ file wakawaka-shakira
wakawaka-shakira: cannot open 'wakawaka-shakira' (No such file or directory)
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP$ file wakawaka-shakira.txt
wakawaka-shakira.txt: cannot open 'wakawaka-shakira.txt' (No such file or directory)
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP$ cd Lab3/
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$ file wakawaka-shakira.txt
wakawaka-shakira.txt: ASCII text
parallel@parallel-Parallels-Virtual-Platform: ~/SISOP/Lab3$
```

Cree 2 archivos con parte de letra de canción usando cat, y luego con el mismo cat, concatene ambos archivos y cree un archivo nuevo con esa concatenación cree un archivo nuevo, el archivo lo cree como .txt y efectivamente me mostró que es ASCII text.

¿Cómo hago copias de seguridad de mis archivos o los copio en un medio de almacenamiento externo?

El comando **cpio** permite copiar archivos en algún medio indicado por el usuario. Así:

cpio opciones <nombrearchivo> rutademedio. Donde las opciones pueden conocerse en el manual, **nombrearchivo** es el archivo que se desea copiar y **rutademedio** es la ruta donde para llegar al medio de almacenamiento deseado. Veamos un ejemplo copiando el archivo practica, creado anteriormente, a una memoria USB:

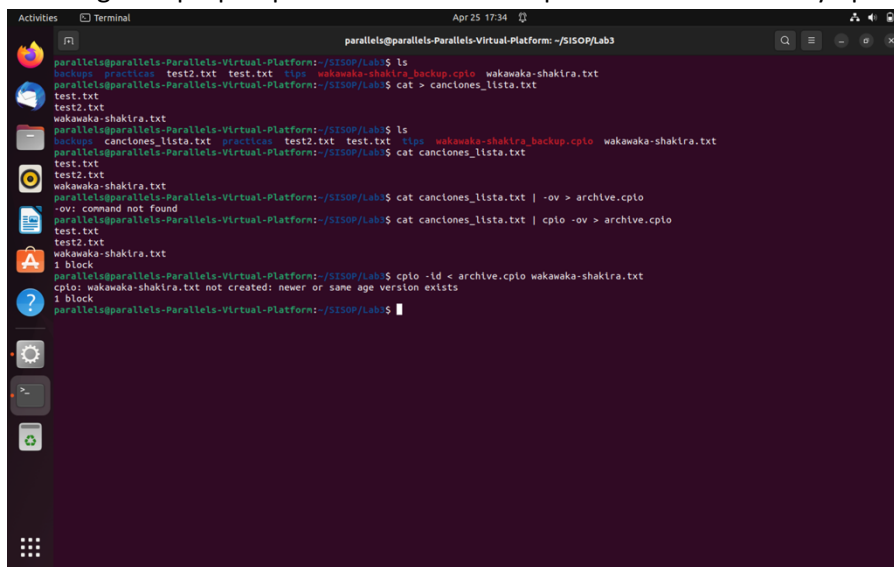
```
cpio -ov <practica> /media/30AD-6631/practica
```

En este caso, /media/30AD-6631 es la ruta creada por el sistema operativo para mi dispositivo USB. La opción -v permite desplegar el nombre de cada archivo copiado.

** Más ejemplos en:

<http://www.mppmu.mpg.de/Introduction/general/subsubsection2.5.8.2.2.html>

4. Haga sus propias pruebas. Describa completamente cada línea y opción utilizada.



```
parallel@parallels-Parallels-Virtual-Platform: ~/SISOP/Lab3
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ ls
backups practicas test2.txt test.txt tips wakawaka-shakira_backup cpio wakawaka-shakira.txt
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ cat > canciones_lista.txt
test.txt
test2.txt
wakawaka-shakira.txt
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ ls
backups canciones_lista.txt practicas test2.txt test.txt tips wakawaka-shakira_backup cpio wakawaka-shakira.txt
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ cat canciones_lista.txt
test.txt
test2.txt
wakawaka-shakira.txt
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ cat canciones_lista.txt | -ov > archive.cpio
-ov: command not found
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ cat canciones_lista.txt | cpio -ov > archive.cpio
test.txt
test2.txt
wakawaka-shakira.txt
1 block
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$ cpio -id < archive.cpio wakawaka-shakira.txt
cpio: wakawaka-shakira.txt not created: newer or same age version exists
1 block
parallel@parallels-Parallels-Virtual-Platform:~/SISOP/Lab3$
```

Cree un .txt listando archivos, y procedí a usar **cat canciones_lista.txt** para luego hacer un **pipe** con el commando **cpio -ov > archive.cpio**:

cpio: es un comando de Linux utilizado para crear y extraer archivos.

-o: esta opción indica que se va a crear un archivo.

-v: esta opción activa la "salida detallada", lo que significa que cpio mostrará información sobre los archivos que está procesando.

>: este símbolo se utiliza para redirigir la salida del comando a un archivo en lugar de a la pantalla.

archive.cpio: este es el nombre del archivo que se creará para contener los archivos o directorios.

En resumen, este comando crea un archivo llamado **archive.cpio** que contiene uno o más archivos o directorios, utilizando el formato de archivo **cpio**.

Luego con el comando **cpio -id < archive.cpio wakawaka-shakira.txt** se utiliza para extraer el archivo **wakawaka-shakira.txt** de un archivo llamado **archive.cpio** que ha sido creado previamente utilizando el formato de archivo **cpio**. **-i** indica que se va a extraer archivos desde el archivo **.cpio**, **-d** indica que se deben crear todos los directorios necesarios para extraer los archivos. Hice este comando en el directorio a propósito para mostrar como el **cpio** no va a extraer **archive** del **.cpio** mientras haya una version de el **archive** a extraer que sea una version igual o mas nueva.

Retroinformación.

1. Entregue cada una de las preguntas de ejercicio.
2. Busque 5 comandos relacionados con los discutidos en esta guía. Pruébelos. Describa sus usos y escriba ejemplos específicos completos, incluyendo la sintaxis y opciones utilizadas.

awk: editor de flujo que realiza manipulación de texto en archivos o entrada estándar.

Ejemplo : **awk '{gsub("cadena_antigua", "cadena_nueva"); print}' archivo.txt > nuevo_archivo.txt**

Este comando **awk** reemplaza todas las apariciones de la cadena "cadena_antigua" por "cadena_nueva" en el archivo **archivo.txt**, y guarda el resultado en un nuevo archivo llamado **nuevo_archivo.txt**. A continuación se explica cada parte del comando:

awk: invoca el comando **awk**.

'{gsub("cadena_antigua", "cadena_nueva"); print}': este es un script **awk** que especifica las acciones a realizar en cada línea del archivo de entrada. **gsub** es una función **awk** que reemplaza todas las

apariciones de un patrón con una cadena de reemplazo. En este caso, estamos reemplazando "**cadena_antigua**" por "**cadena_nueva**". El comando print imprime la línea modificada en la salida estándar.

archivo.txt: especifica el nombre del archivo de entrada que queremos procesar.

'> **nuevo_archivo.txt**': redirige la salida a un nuevo archivo llamado nuevo_archivo.txt. El símbolo > sobrescribe cualquier archivo existente con el mismo nombre, o crea un nuevo archivo si no existe.

Así, cuando se ejecuta este comando, awk lee el archivo de entrada archivo.txt línea por línea, reemplaza todas las apariciones de "**cadena_antigua**" por "**cadena_nueva**", y luego imprime las líneas modificadas en la salida estándar. El símbolo > redirige esta salida a un nuevo archivo llamado **nuevo_archivo.txt**, que contiene la versión modificada del archivo original.

3. ¿En qué situaciones específicas considera que serían útiles los comandos utilizados?

cat: En DevOps, el comando cat se utiliza comúnmente para concatenar archivos de registro o de salida de diferentes fuentes para su posterior análisis. Por ejemplo, se puede utilizar cat para concatenar varios archivos de registro del servidor web para analizarlos y solucionar problemas.

grep: se utiliza comúnmente para buscar patrones en archivos de registro, especialmente en situaciones de depuración. Por ejemplo, si un desarrollador está experimentando problemas con una aplicación en producción, puede utilizar grep para buscar patrones específicos en los registros del servidor para identificar el problema.

cpio: uso para crear archivos de copia de seguridad de bases de datos y otros recursos importantes, para protegerlos de posibles errores humanos o fallos técnicos. Por ejemplo, se puede utilizar cpio para crear un archivo de copia de seguridad de una base de datos crítica y almacenarlo en un sistema de almacenamiento en la nube para su recuperación en caso de un error técnico.

awk: el comando awk se utiliza comúnmente para procesar grandes conjuntos de datos, como archivos de registro de servidores web, registros de aplicaciones y otros archivos similares. Por ejemplo, se puede utilizar awk para extraer datos específicos de un archivo de registro para su posterior análisis, o para filtrar registros en función de ciertos criterios, como la fecha o el tiempo de respuesta.

4. ¿Qué dificultades encontró durante el desarrollo del laboratorio?

No encontré dificultades en sí, pero sí encontré ciertas curiosidades de como los comandos se

comportaban con mi computadora debido a que mi máquina virtual en parte enlazada con los archivos de mi computadora física.

5. ¿Qué mejoraría de esta experiencia de laboratorio?

Considero que para mejorar la experiencia debería agregarse el caso de usar pipe, debido a que los comandos dados en estos laboratorios se complementan y relacionan lo suficiente como para introducir el pipe como herramienta para entrelazar comandos y trabajar de forma más eficiente en linux.

Referencias:

1. Kernighan, B. y Pike, R. El Entorno de programación Unix. Prentice Hall.
2. UTF-8. Disponible en <http://en.wikipedia.org/wiki/UTF-8>

Begginer Linux Tutorial: <http://beginnerlinuxtutorial.com/help-tutorial/basic-linux-commands>