

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

LABORATORIO: Sistemas Operativos

Práctica 4

Archivos y comandos básicos relacionados en Linux

LosDos

Integrantes:

Amanda Velasco Gallardo - 154415
Carlos Octavio Ordaz Bernal - 158525

Fecha(s) de elaboración de la práctica:

15 de febrero de 2019

Introducción

Un sistema de ficheros es la forma en la que el sistema operativo organiza los archivos y directorios que se encuentran dentro del disco duro, de manera que los datos estén de forma estructurada y sin errores. A diferencia de los sistemas de la familia **Windows NT**, dentro de **Linux** no existen unidades físicas, más bien son ficheros que hacen referencia a las unidades.

La estructura de ficheros de **Linux** es jerárquica, formando un árbol cuya raíz es el directorio base o **home**. A partir de éste cuelga toda la estructura del sistema. Para nombrar a los ficheros, directorios y archivos, se puede utilizar cualquier caracter excepto `/`. A la hora de asignar el nombre, **Linux** distingue entre mayúsculas y minúsculas.

Dentro del sistema operativo, existen tres tipos de archivos, los cuales son:

- Archivos ordinarios: contienen la información con la que el usuario trabaja usualmente.
- Directorios: son archivos que contienen referencias a archivos o a otros directorios.
- Archivos especiales: suelen representar dispositivos físicos (*hardware* externo).

Linux distingue tres tipos de usuarios y tres operaciones que los usuarios pueden realizar. Las operaciones son: escritura, lectura y ejecución. En primer lugar, la operación de escritura permite modificar el contenido del archivo. En segundo lugar, la operación de lectura sólo permite visualizar el contenido. Por último, la operación de ejecución permite ejecutar el archivo. Mientras que los tipos de usuarios son: propietario, grupo y otros (el resto de los usuarios). El usuario propietario, o simplemente usuario, es aquel que genera o crea el archivo dentro de un directorio de trabajo o dentro del directorio base. Un grupo de trabajo, es un conjunto de usuarios que tienen los mismos privilegios dentro del sistema.

Mediante el uso del comando **chmod** es posible cambiar los permisos que cada uno de los tipos de usuario puede tener sobre un determinado archivo o directorio. Los permisos se pueden especificar de manera explícita por medio de las letras *u*, *g*, y *o* para los tres tipos de usuarios, y *r*, *w* y *x* para las tres operaciones. Suponiendo que se desean asignar todos los permisos al usuario

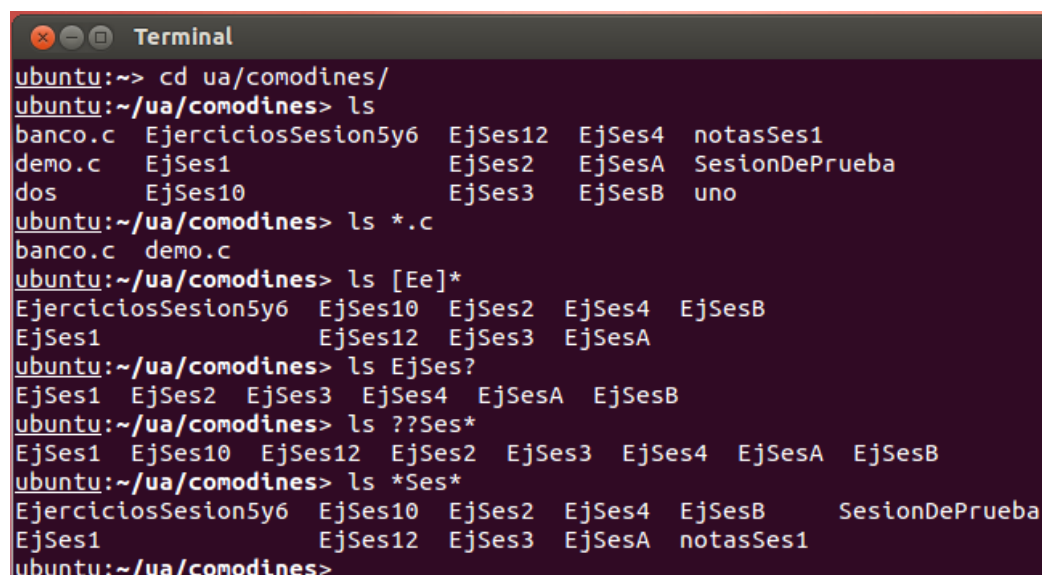
propietario, y sólo lectura a los otros, el comando sería: `chmod u+rw, o+r archivo`. Otra forma es especificar los permisos en modo octal, donde se representan en tres dígitos (uno por tipo de usuario); el 1 es sólo permiso de ejecución, el 2 sólo permiso de escritura, y el 4 es sólo permiso de lectura. Así, la suma de los permisos para cada dígito es la expresión del permiso, de manera similar al ejemplo anterior el comando sería: `chmod 744 archivo`.

Desarrollo

2. Pasar al subdirectorio *comodines* dentro del subdirectorio *ua*.

- *Listar todos los archivos que tengan el patrón “Ses” en los caracteres 3, 4 y 5 de su nombre:* Se utilizó el comando: `ls ??Ses*`, para cumplir lo solicitado.
- *Listar todos los archivos que tengan el patrón “Ses” en cualquier parte del nombre:* Para listar los archivos pedidos se usó el comando: `ls *Ses*`.

La siguiente figura 1 muestra que se hicieron las pruebas sugeridas en el archivo *UbuntuCoArchi.pptx* y además el resultado de los dos puntos anteriores.



```

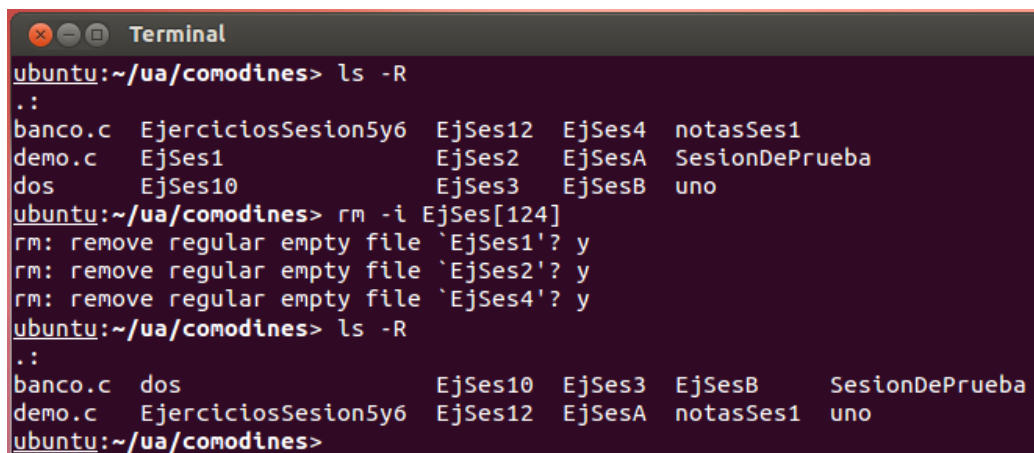
Terminal
ubuntu:~> cd ua/comodines/
ubuntu:~/ua/comodines> ls
banco.c  EjerciciosSesion5y6  EjSes12  EjSes4  notasSes1
demo.c   EjSes1               EjSes2   EjSesA  SesionDePrueba
dos      EjSes10              EjSes3   EjSesB  uno
ubuntu:~/ua/comodines> ls *.c
banco.c  demo.c
ubuntu:~/ua/comodines> ls [Ee]*
EjerciciosSesion5y6  EjSes10  EjSes2  EjSes4  EjSesB
EjSes1              EjSes12  EjSes3  EjSesA
ubuntu:~/ua/comodines> ls EjSes?
EjSes1  EjSes2  EjSes3  EjSes4  EjSesA  EjSesB
ubuntu:~/ua/comodines> ls ??Ses*
EjSes1  EjSes10  EjSes12  EjSes2  EjSes3  EjSes4  EjSesA  EjSesB
ubuntu:~/ua/comodines> ls *Ses*
EjerciciosSesion5y6  EjSes10  EjSes2  EjSes4  EjSesB  SesionDePrueba
EjSes1              EjSes12  EjSes3  EjSesA  notasSes1
ubuntu:~/ua/comodines>

```

Fig. 1: Resultado de los comandos para listar archivos.

3. Con un solo comando y utilizando caracteres comodines borrar los archivos *EjSes1* *EjSes2* y *EjSes4*. El comando debe solicitar confirmación antes de borrar un archivo.

La figura 2 muestra el contenido inicial del directorio, la ejecución del comando para borrar los archivos solicitados y el directorio después de ingresar el comando.



```

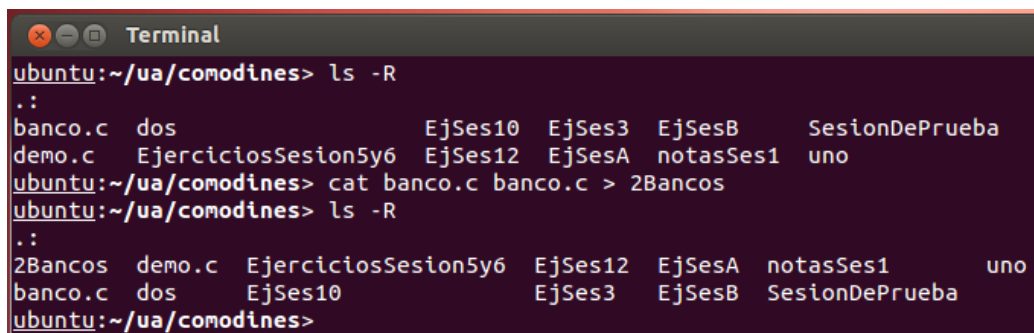
Terminal
ubuntu:~/ua/comodines> ls -R
.:
banco.c  EjerciciosSesion5y6  EjSes12  EjSes4  notasSes1
demo.c   EjSes1                EjSes2   EjSesA  SesionDePrueba
dos      EjSes10               EjSes3   EjSesB  uno
ubuntu:~/ua/comodines> rm -i EjSes[124]
rm: remove regular empty file `EjSes1'? y
rm: remove regular empty file `EjSes2'? y
rm: remove regular empty file `EjSes4'? y
ubuntu:~/ua/comodines> ls -R
.:
banco.c  dos                EjSes10  EjSes3  EjSesB  SesionDePrueba
demo.c   EjerciciosSesion5y6  EjSes12  EjSesA  notasSes1  uno
ubuntu:~/ua/comodines>

```

Fig. 2: Resultado del comando para borrar archivos.

4. Con un solo comando, generar un archivo *2Bancos* que tenga dos copias del archivo *banco.c*.

La siguiente figura 3 muestra el comando *cat* utilizado para crear dos copias del archivo en uno nuevo. Además se lista el directorio antes y después de la ejecución del comando para comprobar el resultado.



```

Terminal
ubuntu:~/ua/comodines> ls -R
.:
banco.c  dos                EjSes10  EjSes3  EjSesB  SesionDePrueba
demo.c   EjerciciosSesion5y6  EjSes12  EjSesA  notasSes1  uno
ubuntu:~/ua/comodines> cat banco.c banco.c > 2Bancos
ubuntu:~/ua/comodines> ls -R
.:
2Bancos  demo.c  EjerciciosSesion5y6  EjSes12  EjSesA  notasSes1  uno
banco.c  dos     EjSes10              EjSes3   EjSesB   SesionDePrueba
ubuntu:~/ua/comodines>

```

Fig. 3: Resultado del comando para generar el archivo *2Bancos*.

5. Revisar los permisos de 2Bancos y después ejecutar los comandos para que:

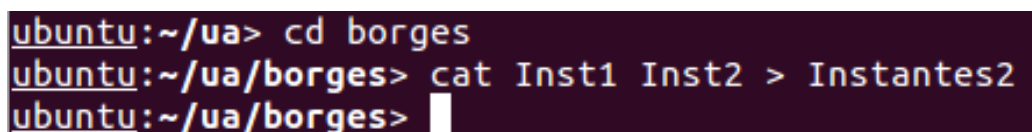
- a) *El usuario tenga todos los permisos. Los otros solo de lectura: `chmod u+rwx, o=r 2Bancos`.*
- b) *Nadie pueda escribir el archivo: `chmod a-w 2Bancos`.*
- c) *Los otros puedan únicamente ejecutar: `chmod o=x 2Bancos`.*
- d) *El usuario tenga todos los permisos. El grupo sólo de lectura y ejecución y los otros únicamente de ejecución: `chmod u+rwx, g=r, o=x 2Bancos`.*
- e) *El grupo tenga los mismos permisos que los otros: `chmod g=o 2Bancos`.*

Ahora borrar todos los permisos de todos, esto se hizo con el comando `chmod a= 2Bancos`. Y repetir los incisos *a - d* en modo octal.

- a) *El usuario tenga todos los permisos. Los otros solo de lectura: `chmod 704 2Bancos`.*
- b) *Nadie pueda escribir el archivo: `chmod 555 2Bancos`.*
- c) *Los otros puedan únicamente ejecutar: `chmod 771 2Bancos`.*
- d) *El usuario tenga todos los permisos. El grupo sólo de lectura y ejecución y los otros únicamente de ejecución: `chmod 751 2Bancos`.*

6. Pasar al subdirectorio *borges* y generar un archivo *Instantes2* que sea la concatenación de los archivos *Inst1* e *Inst2*.

La figura 4 muestra el uso del comando `cat` para generar el archivo *Instantes2* que contiene a los archivos *Inst1* e *Inst2*.

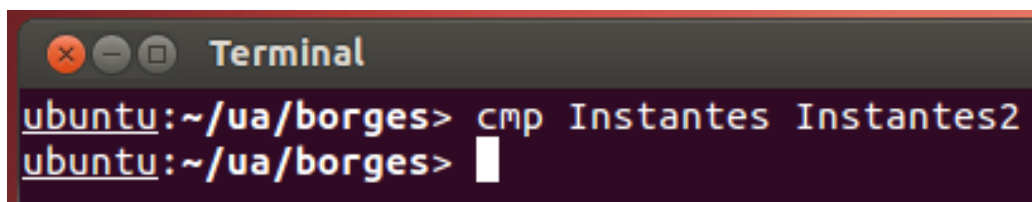


```
ubuntu:~/ua> cd borges
ubuntu:~/ua/borges> cat Inst1 Inst2 > Instantes2
ubuntu:~/ua/borges> █
```

Fig. 4: Resultado del comando para generar el archivo *Instantes2*.

7. Verificar que el archivo creado es exactamente igual al archivo *Instantes*.

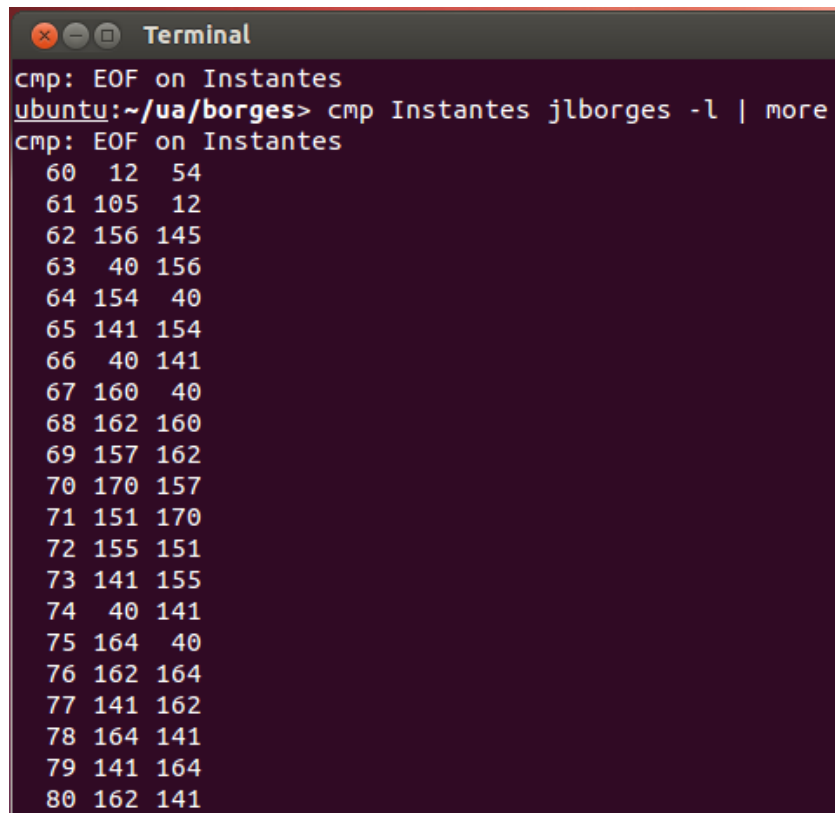
La siguiente figura 5 permite ver que, al usar el comando `cmp` entre los dos archivos, el resultado no mostró ninguna línea por lo que se infiere que los archivos son iguales.

A terminal window titled "Terminal" with standard window controls (close, minimize, maximize). The prompt is "ubuntu:~/ua/borges>". The command "cmp Instantes Instantes2" has been entered. The output is a blank line, indicating that the files are identical.

```
ubuntu:~/ua/borges> cmp Instantes Instantes2
ubuntu:~/ua/borges>
```

Fig. 5: Resultado del comando para comprobar que los archivos son iguales.

8. Comparar el archivo *Instantes* con el archivo *jlborges* e identificar todas las diferencias que existan entre estos dos archivos. La figura 6 muestra las diferencias entre los dos archivos. Se especifica en la lista cada byte que difiere, indicando su posición y su valor en octal.



```
Terminal
cmp: EOF on Instantes
ubuntu:~/ua/borges> cmp Instantes jlborges -l | more
cmp: EOF on Instantes
 60  12  54
 61 105  12
 62 156 145
 63  40 156
 64 154  40
 65 141 154
 66  40 141
 67 160  40
 68 162 160
 69 157 162
 70 170 157
 71 151 170
 72 155 151
 73 141 155
 74  40 141
 75 164  40
 76 162 164
 77 141 162
 78 164 141
 79 141 164
 80 162 141
```

Fig. 6: Resultado del comando para mostrar las diferencias entre los dos archivos.

9. ¿Qué tipo de archivo son los archivos encontrados en el directorio *comodines*? ¿En el directorio base?

De las mostradas a continuación, en la primer imagen podemos identificar que en el directorio *comodines* encontramos archivos del tipo texto en formato ASCII y algunos otros vacíos.

```
ubuntu:~/ua> cd comodines
ubuntu:~/ua/comodines> ls > 9.txt
ubuntu:~/ua/comodines> file -f 9.txt
2Bancos:          ASCII C program text
9.txt:            ASCII text
banco.c:          ASCII text
demo.c:           empty
dos:              ASCII text
EjerciciosSesion5y6: empty
EjSes10:          empty
EjSes12:          empty
EjSes3:           empty
EjSesA:           empty
EjSesB:           empty
notasSes1:        empty
SesionDePrueba:   empty
uno:              ASCII text
ubuntu:~/ua/comodines>
```

Fig. 7: Resultado del comando para mostrar los tipos de archivo en el directorio *comodines*.

En la segunda, podemos encontrar archivos del tipo texto en formato ASCII, documentos HTML y directorios dentro del directorio base.


```
ubuntu:~/ua/comodines> cd
ubuntu:~> ls > 9b.txt
ubuntu:~> file -f 9b.txt
12.txt:          ASCII text
9b.txt:          ASCII text
Desktop:        directory
Documents:      directory
Downloads:      directory
examples.desktop: UTF-8 Unicode text
index.html:     HTML document, ASCII text
index.html.1:   HTML document, ASCII text
mod1:           directory
mod2:           directory
mod3:           directory
Music:          directory
Pictures:       directory
Public:         directory
Templates:      directory
ua:             directory
Videos:         directory
workspace:      directory
ubuntu:~> 
```

Fig. 8: Resultado del comando para mostrar los tipos de archivo en el directorio base.

10. Ejecutar la siguiente secuencia y explicar cómo funciona: `keops>cd~/; ls -l >FileArch; file -f FileArch | more.` ¿Por qué?

Primero se cambia al directorio base con `cd~/`. Al aplicar `ls`, con el `-l` se devuelve una lista con un archivo por línea, en cambio el `-l` muestra el formato largo con detalles de los archivos, lo cual no puede ser leído bien por el `file -f`. El resultado del comando se muestra en la siguiente figura 9.

```
ubuntu:~> cd ~ ; ls -l > FileArch ; file -f FileArch | more
12.txt:      ASCII text
9b.txt:      ASCII text
Desktop:     directory
Documents:   directory
Downloads:   directory
examples.desktop: UTF-8 Unicode text
FileArch:    ASCII text
index.html:  HTML document, ASCII text
index.html.1: HTML document, ASCII text
mod1:        directory
mod2:        directory
mod3:        directory
Music:       directory
Pictures:    directory
Public:      directory
Templates:   directory
ua:          directory
Videos:      directory
workspace:   directory
ubuntu:~>
```

Fig. 9: Resultado del comando.

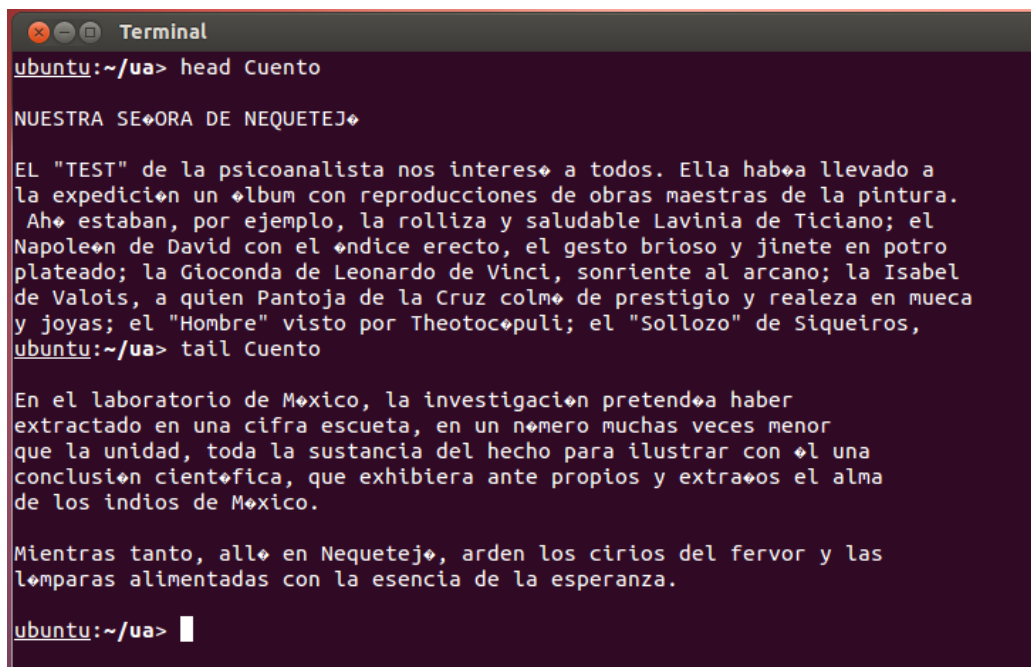
11. Mover el archivo *FileArch* al subdirectorio *borges*.

El comando `mv` se utilizó para cambiar la ubicación del archivo al directorio solicitado. La siguiente figura 10 muestra el resultado.

```
ubuntu:~> mv FileArch ua/borges/
ubuntu:~> cd ua/borges
ubuntu:~/ua/borges> ls
cortazar1 cortazar2 FileArch Inst1 Inst2 Instantes Instantes2 jlborges
ubuntu:~/ua/borges>
```

Fig. 10: Resultado del comando para mover el archivo a un directorio específico.

12. Despliegue el archivo *Cuento* con los filtros `head`, `tail` y después con el paginador `more`. De las mostradas a continuación, en la primer imagen podemos identificar que sólo se pueden ver las primeras o últimas líneas del archivo.



```
Terminal
ubuntu:~/ua> head Cuento

NUESTRA SEÑORA DE NEQUETEJO

EL "TEST" de la psicoanalista nos interesó a todos. Ella había llevado a
la expedición un álbum con reproducciones de obras maestras de la pintura.
Ahí estaban, por ejemplo, la rolliza y saludable Lavinia de Ticiano; el
Napoleón de David con el índice erecto, el gesto brioso y jinete en potro
plateado; la Gioconda de Leonardo de Vinci, sonriente al arcano; la Isabel
de Valois, a quien Pantoja de la Cruz colmó de prestigio y realeza en mueca
y joyas; el "Hombre" visto por Theotocópuli; el "Sollozo" de Siqueiros,
ubuntu:~/ua> tail Cuento

En el laboratorio de México, la investigación pretendía haber
extractado en una cifra escueta, en un número muchas veces menor
que la unidad, toda la sustancia del hecho para ilustrar con él una
conclusión científica, que exhibiera ante propios y extraños el alma
de los indios de México.

Mientras tanto, allá en Nequetejo, arden los cirios del fervor y las
lámparas alimentadas con la esencia de la esperanza.

ubuntu:~/ua> █
```

Fig. 11: Resultado del comandos para mostrar las primeras y últimas líneas del archivo.

En la segunda, podemos ver que el contenido del archivo se mostró con el paginador.

```
ubuntu:~/ua> cat Cuento | more

NUESTRA SEÑORA DE NEQUETEJO

EL "TEST" de la psicoanalista nos interesó a todos. Ella había llevado a
la expedición un álbum con reproducciones de obras maestras de la pintura.
Ahí estaban, por ejemplo, la rolliza y saludable Lavinia de Ticiano; el
Napoleón de David con el índice erecto, el gesto brioso y jinete en potro
plateado; la Gioconda de Leonardo de Vinci, sonriente al arcano; la Isabel
de Valois, a quien Pantoja de la Cruz colmó de prestigio y realeza en mueca
y joyas; el "Hombre" visto por Theotocopuli; el "Sollozo" de Siqueiros,
donde la mujer empuja el dolor en escalofriante actitud; el patético "Tata
Jesucristo" de Goitia; el "Zapata" de Diego, santón bigotudo, baqueano de
hambrientos y portaestandarte de causas albeantes como los calzones
blancos y la blanca sonrisa de los indios; la "Trinchera", encrucijada de
tragedia y nidal de maldiciones, en que José Clemente Orozco vació la
intención en forma y erigió la protesta en colores y, en fin...
Los indígenas de aquel lugarejo Nequetejo, de aquella aldehuela perdida
en las rugosidades de la Sierra Madre, miraban y miraban con admiración
callada las lóminas que despertaban en ellos excelencias y calidades
agazapadas entre el moho de sus afrentas y el humazo de sus recelos. La vista
punzante sobre los cromos y en las pupilas dilatadas por el pasmo, las gamas,
los tonos y las formas reflejadas con la misma saña, con la misma furia con
que el impacto estético había lesionado más los corazones que los cerebros.
--More--
```

Fig. 12: Resultado del comando para mostrar el archivo con el paginador.

13. Agregar su nombre al final del archivo *Cuento* con un solo comando de los que hemos visto hasta ahora.

La figura 13 muestra los comandos utilizados para añadir los nombres hasta el final del archivo.

```
ubuntu:~/ua> echo "Amanda" >> Cuento
ubuntu:~/ua> echo "Octavio" >> Cuento
ubuntu:~/ua> █
```

Fig. 13: Comandos utilizados para agregar los nombres.

14. Cambiar la máscara de usuario *umask* para inhibir los permisos de escritura para todos y de ejecución para el grupo y los otros. Generar archivos y directorios con los mecanismos anteriores y comprobar con qué permisos estos archivos han sido creados.

El comando `umask 233` inhibió los siguientes permisos a todos los archivos del directorio: `--w--wx-wx` (al usuario de escritura, y a los demás de

escritura y ejecución). En la siguiente figura podemos observar que se modificaron los permisos, y con el comando `touch` comprobamos que sí hubo cambios.

```
ubuntu:~/ua> umask 233
ubuntu:~/ua> umask
233
ubuntu:~/ua> ls -l
total 48
drwxr-xr-x 2 sisops sisops 4096 Feb 15 09:26 borges
drwxr-xr-x 2 sisops sisops 4096 Feb 15 09:18 comodines
-rw-r--r-- 1 sisops sisops 14388 Feb 15 09:39 Cuento
-rw-r--r-- 1 sisops sisops 14390 Feb 15 09:37 Cuento~
drwxr-xr-x 2 sisops sisops 4096 Feb 15 07:28 f-rojas
-rw-rw-r-- 1 sisops sisops 7 Feb 15 09:34 nombres.txt~
ubuntu:~/ua> touch nombres
ubuntu:~/ua> touch nombres.txt
ubuntu:~/ua> ls -l
total 48
drwxr-xr-x 2 sisops sisops 4096 Feb 15 09:26 borges
drwxr-xr-x 2 sisops sisops 4096 Feb 15 09:18 comodines
-rw-r--r-- 1 sisops sisops 14388 Feb 15 09:39 Cuento
-rw-r--r-- 1 sisops sisops 14390 Feb 15 09:37 Cuento~
drwxr-xr-x 2 sisops sisops 4096 Feb 15 07:28 f-rojas
-r--r--r-- 1 sisops sisops 0 Feb 15 09:46 nombres
-r--r--r-- 1 sisops sisops 0 Feb 15 09:46 nombres.txt
-rw-rw-r-- 1 sisops sisops 7 Feb 15 09:34 nombres.txt~
ubuntu:~/ua>
```

Fig. 14: Comandos utilizados para agregar los nombres.

15. Ejecutar el siguiente comando desde el directorio base: `prompt>(pwd; ls -RF1;) >XXXXp04.txt`, donde **XXXX** es su clave única.

El resultado de este comando se incluye en un archivo de texto adjunto a este reporte. El comando mostró el contenido de cada uno de los directorios existentes especificando: los permisos, tipo de archivo, fecha y hora de creación, nombre del usuario, y nombre del archivo.

Conclusiones

Esta práctica nos permitió profundizar en mayor medida en el uso de comandos dentro de la terminal de **Linux**. De manera específica, se estudiaron las distintas opciones con las que se puede utilizar el comando **ls** para poder listar todo el contenido de un directorio. Además, se utilizaron comandos como **cat**, **mv**, **cp**, **rm** para crear, mover, o modificar archivos. Por último, aprendimos dos maneras (símbolica u octal) para modificar los permisos de un archivo dentro de un directorio en específico. Esto es de gran importancia porque hay ocasiones en las que es necesario limitar las acciones que un usuario, o grupos de usuarios, puedan realizar sobre de un archivo. En el caso de tener que modificar todos los permisos dentro de un directorio se puede utilizar el comando **umask** para lograrlo, éste también se implementó en el desarrollo de la práctica.

Referencias

- (2012). Permisos y derechos en Linux. Recuperado el 19 de febrero de 2019, del sitio web: <https://blog.desdelinux.net/permisos-y-derechos-en-linux/>
- Huguet, O. Linux. Sistema de Ficheros. Recuperado el 19 de febrero del 2019, del sitio web: http://mural.uv.es/oshuso/823_directorios_ms_importantes_en_linux.html
- Ríos, J. (2019). Notas del curso de Sistemas Operativos. Recuperado el 19 de febrero de 2019, del sitio web: Comunidad ITAM.