Facilitadora: Prof. Ahlam Almaeni Asignatura: Sistemas Operativos I

Estudiante: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Grupo: 11R222

Estudiante: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **TÍTULO** **DE LA EXPERIENCIA**: COMANDOS PARA EL MANEJO DE ARCHIVOS.
2. **TEMAS:**

Utilización de comandos básicos para el manejo de archivos.

1. **OBJETIVO(S):**
   * Explorar el uso de comandos que permitan la manipulación de archivos en ambiente Linux.
2. **RECURSOS:**

* Un computador personal.
* SO Linux Fedora.
* Guía del laboratorio.

1. **RÚBRICAS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterios** | **2** | **1** | **0** |
| I – Identificación del/los participante/s | 100% | El 50% | Ninguno |
| II - Proceso – Utilizó los recursos recomendados en el enunciado o procedimiento. | 100% | Más del 50% | Menos del  50% |
| III - Solución – Presentó los datos solicitados | 100% | Entre 50% y  70%. | Menos del  50% |
| IV – Puntualidad en la entrega | 100% | Entregó después de la fecha | No entregó |
| V - Formato – Siguió el formato presentado. | 100% | Obvió algunos puntos | No siguió el formato |

1. **ENUNCIADO DE LA EXPERIENCIA O PROCEDIMIENTO:**

* Desarrolle el laboratorio en **grupo de 2**.
* Lea detenidamente la guía completa antes de iniciarla.
* Practique cada uno de los comandos en la sección de procedimiento.
* Capture las pantallas generadas con cada ejecución de comandos y colóquelas inmediatamente después del comando ejecutado. Respuestas de texto escribirlas en **rojo**.

**Procedimiento:**

**El comando *cat***

El comando **cat** sirve para concatenar varios archivos que posteriormente se desplegarán en pantalla. Admite como argumentos la lista de archivos que hay que enlazar. Su sintaxis es la siguiente:

|  |  |
| --- | --- |
|  | $ cat -opciones archivos |

* **Argumentos**

La instrucción **cat** admite como argumentos una lista de archivos que hay que enlazar. Estos se encadenarán en el mismo orden en el cual fueron especificados. Por ejemplo, suponga que posee los siguientes archivos:

* **Archivo A**cuyo contenido es *Esta es la primera línea*
* **Archivo B** cuyo contenido es *Esta es la segunda línea*

Si se quisiera enlazar los archivos, sería de la siguiente manera:

|  |  |
| --- | --- |
|  | $ cat A B |

Obteniendo el siguiente resultado:

|  |  |
| --- | --- |
|  | 1. Esta es la primera línea 2. Esta es la segunda línea |

Si sólo se llegase a especificar un sólo archivo, el resultado del comando sería el contenido del archivo mismo. Ejemplo:

|  |  |
| --- | --- |
|  | $ cat A |

Desplegando como resultado:

|  |  |
| --- | --- |
|  | Esta es la primera línea |

Si no se especifica ningún argumento, **cat** tomará el contenido desde el teclado, desplegando la información que el usuario ingrese, hasta que este presione **Ctrl D**.

* **Opciones**

**-n:** Numera las líneas mostradas (Desde 1).

Ejemplo:

|  |  |
| --- | --- |
|  | $cat -n /etc/hosts ~/.bashrc |

**-b:** Numera las líneas mostradas, pero no se numeran las líneas vacías (en blanco).

Ejemplo:

|  |  |
| --- | --- |
|  | $cat -s /etc/shells /etc/passwd |

**-s:**Suprime las múltiples líneas vacías contiguas a una sola línea en blanco. Ejemplo:

|  |  |
| --- | --- |
|  | $cat -s /etc/hosts /etc/passwd |

**El comando *file***

**file** determina el tipo de un archivo y te imprime en pantalla el resultado. No hace falta que el archivo tenga una extensión para que File determine su tipo, pues la aplicación ejecuta una serie de pruebas sobre el mismo para tratar de clasificarlo.

Ejecutarlo sería tan sencillo como:

$ file un\_archivo\_de\_texto.txt

**El comando *ln***

**ln** es una aplicación que permite crear enlaces a los archivos, tanto físicos (*hard links*) como simbólicos (*soft links*). En pocas palabras, un enlace simbólico es como un *acceso directo* en Windows o un *alias* en OSX mientras que un enlace físico es un nombre diferente para la misma información en disco.

Para crear un enlace físico ejecutamos:

$ln archivo\_origen nombre\_enlace

Y para crear un enlace simbólico:

$ln -s archivo\_origen nombre\_enlace

**El comando *cmp***

**cmp** compara el contenido de dos archivos y devuelve 0 si los archivos son idénticos ó 1 si los archivos tienen diferencias. En caso de error devuelve -1.

Para ejecutarlo basta con:

$ cmp -s archivo1 archivo2

**cmp** también puede mostrar algo de información sobre las diferencias, pero para un reporte más detallado tenemos el siguiente comando.

**El comando *diff***

**diff**, al igual que cmp, compara el contenido de dos archivos, pero en lugar de devolver un valor imprime en pantalla un resumen detallado línea a línea de las diferencias. Ejecutarlo es tan simple como:

$ diff archivo1.txt archivo2.txt

**diff** también puede usarse con directorios. En este caso comparará los nombres de los archivos correspondientes en cada directorio por orden alfabético e imprimirá en pantalla los archivos que estén en un directorio, pero no estén en el otro.

**El comando *wc***

wc imprime en pantalla la cantidad de saltos de línea, palabras y bytes totales que contenga un archivo. Para usarlo con un archivo cualquiera ejecutamos:

$wc archivo.txt

**El comando *sort***

***sort*** imprime en pantalla las líneas de un archivo ordenadas alfabéticamente. Para ejecutarlo basta con:

$sort archivo.txt

**El comando *tail***

Tail muestra en pantalla las últimas líneas de un archivo.

$tail archivo.txt

Por defecto siempre muestra 10 pero podemos indicarle un número diferente de líneas a visualizar usando el parámetro -n:

$tail -n 50 archivo.txt

**El comando *head***

**head** es el comando opuesto a **tail**, muestra las primeras líneas de un archivo.

$head archivo.txt

Al igual que **tail**, muestra por defecto las 10 primeras líneas, pero podemos indicarle un número diferente usando el parámetro -n:

$head -n archivo.txt

**El comando *more***

**more** es un filtro que permite paginar el contenido de un archivo para que se vea a razón de una pantalla a la vez. Era muy usado en las viejas terminales cuya resolución era de 80×25 para visualizar archivos muy grandes. Para utilizarlo simplemente ejecutamos:

$more archivo.txt

También permite navegar a través del contenido del archivo usando las flechas direccionales *arriba* y *abajo*, *Espacio* o la tecla *Enter*. Para salir de more usamos la tecla *Q.*

**El comando *less***

Aunque su nombre es lo opuesto de **more** es realmente una versión mejorada de éste último.

**less** es otro filtro que permite paginar el contenido de un archivo pero que además de permitir la navegación hacia adelante y hacia atrás, está optimizado para trabajar con archivos muy grandes.

Ejecutarlo es tan simple como escribir:

$less archivo.txt

**less** permite navegar a través del contenido del archivo usando las flechas direccionales *arriba* y *abajo*, *Espacio* o la tecla *Enter*.

Para salir de *less* también usamos la tecla *Q*.

Recuerda que para obtener más información sobre los parámetros y la sintaxis de los comandos puedes usar la aplicación **man** desde la terminal. Por ejemplo:

$man less