MANIPULACIÓN DE ARCHIVOS DE TEXTO.

Ingrese a un shell de línea de comandos de su equipo .

Determinando el tipo de un archivo
En muchas ocasiones es extremadamente útil determinar que clase de contenido tiene un archivo antes de utilizarlo. Para determinar el tipo de un archivo determinado se utiliza el comando **`file`**.

Sintaxis: **`file <opciones> <archivo>`**

- Ingrese el comando **`file --help`** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando file?
- Ingrese el comando **`file /etc/passwd`** y a continuación <ENTER>. ¿De qué tipo de archivo se trata?
- Ingrese el comando **`file /bin/bash`** y a continuación <ENTER>. ¿Y este archivo?
- Ingrese el comando **`file /dev/tty1`** y a continuación <ENTER>. ¿Qué tipo de archivo es?
- ¿De qué tipo es el archivo **`/dev/sda`**? ¿Y el archivo **`/home`**? ¿Y el archivo /**`dev/psaux`**?

Otra manera de determinar el tipo de archivo es mediante la opción de mostrar los archivos en colores del comando ls (**`ls --color`**, vista en el Laboratorio 1).

Mostrando por pantalla el contenido de un archivo de texto
En incontables ocasiones el usuario se verá en la necesidad de ver por pantalla de manera rápida y simple el contenido de un archivo de texto dado.

Una de las posibles alternativas es el uso del comando **`more`**. El comando more tiene la finalidad de mostrar el contenido de un archivo de texto por pantalla de manera paginada (deteniéndose cada vez que se llena la pantalla).

Sintaxis: **`more <opciones> <archivo>`**

- Ingrese el comando **`more /etc/hosts`** y a continuación <ENTER>. ¿Ve el contenido del archivo?
- Ahora Ingrese el comando **`more /etc/services`** y a continuación <ENTER>. ¿Ve el contenido del archivo? ¿En qué se diferencia el comportamiento del comando more?
- Presione la barra espaciadora. ¿Cuánto se desplazo el texto?
- Presione la tecla <ENTER>. ¿Cuánto se desplazo el texto ahora?

Otra manera de ver el contenido de un archivo de texto es el uso del comando **`less`**. El comando less tiene la finalidad de mostrar el contenido de un archivo de texto por pantalla de manera paginada (deteniéndose cada vez que se llena la pantalla). El comando less permite ver la pantalla siguiente con las mismas teclas que more y, si lo deseamos, a diferencia de more, podemos ver la pantalla anterior usando las teclas del cursor o bien mediante **`<b>`** e **`<y>`**.

- Ahora Ingrese el comando **`less /etc/services`** y a continuación <ENTER>. ¿En qué se diferencia del comportamiento del comando more?

Mostrando por pantalla sólo parte del contenido de un archivo de texto
Cuando trabajemos con archivos de texto cuyo tamaño sea considerable (esto es algo realmente común en las tareas habituales de un administrador de servidores Linux) en ocasiones sólo deseamos ver parte de un archivo de texto (usualmente las *n* primeras líneas o las **n** últimas líneas).

Visualizando el comienzo de un archivo
Para visualizar las n primeras líneas de un archivo de texto se utiliza el comando **head**.

Sintaxis: **head *<opciones> <archivo>***

- Ingrese el comando **head --help** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando head?
- Ingrese el comando **head /etc/services** y a continuación <ENTER>. ¿Cuántas líneas del archivo /etc/services le mostró?
- Ahora Ingrese el comando **head -n 15 /etc/services** y a continuación <ENTER>. ¿Cuántas líneas del archivo /etc/services le mostró ahora?

Visualizando el final de un archivo
Para visualizar las n últimas líneas de un archivo de texto se utiliza el comando **tail**.

Sintaxis: **tail *<opciones> <archivo>***

- Ingrese el comando **tail --help** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando tail?
- Ingrese el comando **tail /etc/services** y a continuación <ENTER>. ¿Cuántas líneas del archivo **/etc/services** le mostró?
- Ahora Ingrese el comando **tail -n 18 /etc/services** y a continuación <ENTER>. ¿Cuántas líneas del archivo **/etc/services** le mostró ahora?

Buscando patrones dentro de un archivo de texto
Muchas veces necesitamos encontrar las líneas de un archivo de texto que contienen una determinada subcadena o patrón de texto. Para mostrar las líneas que contienen un patrón de texto se utiliza el comando **grep**.

Sintaxis: **grep *<opciones> <patrón> <archivo>***

- Ingrese el comando **grep --help** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando grep?
- Ingrese el comando **grep alumno /etc/passwd** y a continuación <ENTER>. ¿Qué haría este comando?¿Qué información mostró?¿Porqué mostró solamente la línea alumno?
- Ingrese el comando **grep bash /etc/passwd** y a continuación <ENTER>. ¿Qué haría este comando?¿Qué información mostró?
- Ahora el comando **grep –v bash /etc/passwd** y a continuación <ENTER>. ¿Qué haría este comando?¿Qué información mostró?¿Nota la diferencia?

Contando las líneas/palabras/bytes de un archivo
Para contar cuántas líneas/palabras/bytes tiene un archivo de texto se utiliza el comando **wc**.

Sintaxis: ***wc <opciones> <archivo>***

- Ingrese el comando **wc --help** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando wc?
- Ingrese el comando **wc /etc/passwd** y a continuación <ENTER>. ¿Qué haría este comando? ¿Qué información mostró sobre el archivo /etc/passwd?
- Ingrese el comando **wc -l /etc/passwd** y a continuación <ENTER>. ¿Qué información mostró sobre el archivo /etc/passwd?
- Ingrese el comando **wc -c /etc/passwd** y a continuación <ENTER>. ¿Qué información mostró sobre el archivo /etc/passwd?

- Ingrese el comando **`wc -w /etc/passwd`** y a continuación <ENTER>. ¿Qué información mostró sobre el archivo /etc/passwd?

Separando (partiendo) un archivo

Para partir un archivo en dos o mas partes de igual tamaño se emplea el comado **`split`**.

Sintaxis: ***`split <opciones> <archivo> <prefijo>`***

- Ingrese el comando **`split --help`** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando split?
- Ingrese el comando **`split -l 10 /etc/passwd partido`** y a continuación <ENTER>. ¿Qué haría este comando? Liste los archivos del directorio actual. ¿En cuantas partes separo el archivo password? Determine cuántas líneas tiene cada parte usando el comando wc. Borre todas las partes del archivo **`partido`**.
- Ingrese el comando **`split -b 10 /etc/passwd partido`** y a continuación <ENTER>. ¿Qué haría este comando?Liste los archivos del directorio actual. ¿En cuántas partes separó el archivo password? ¿Son más que las de la opción **`-l 10`**, por qué? Borre todas las partes del archivo **`partido`**.
- Ingrese el comando **`split -b 10k /etc/passwd partido`** y a continuación <ENTER>. ¿Qué haría este comando?Liste los archivos del directorio actual. ¿En cuántas partes separó el archivo password? ¿Son más o menos que las de la opción **`-b 10`**?¿ por qué? Borre todas las partes del archivo **`partido`**.

Uniendo (concatenando) varios archivos

Para concatenar uno o más archivos y visualizarlos en la salida estándar (pantalla) se utiliza el comando **`cat`**.

Sintaxis: ***`cat <opciones> <archivo1> <archivo2> ... <archivon>`***

- Ingrese el comando **`cat --help`** y a continuación <ENTER>. ¿Para qué sirven las opciones del comando cat?
- Ingrese el comando **`split -l 10 /etc/passwd partido`** y a continuación <ENTER>.
- Ingrese el comando **`cat partidoaa`** y a continuación <ENTER>. ¿Que ocurrío?
- Ingrese el comando **`cat partidoaa partidoab`** y a continuación <ENTER>. ¿Y ahora? ¿Nota cómo funciona el comando cat?

El editor Vi

Vi (pronunciado algunas veces como vee-eye, es una abreviatura de "visual") provee capacidad de edición de texto básicas. Hay tres aspectos de vi que lo hacen atractivo. Primero, vi es proporcionado con todos los sistemas UNIX . Ud. puede usar vi en cualquier lugar donde exista un sistema operativo Unix. Segundo, vi usa una pequeña cantidad de memoria, lo cual permite una operación eficiente cuando la red está muy ocupada. Tercero, porque vi usa teclas alfanuméricas estándares para los comandos, usted puede usarlo en alguna terminal virtual o una estación de trabajo sin tener que preocuparse por mapeos de teclas inusuales.

Todas las distribuciones de linux incluyen una versión del editor Vi... Para invocar al editor Vi:

Sintaxis: **`vi`** ***`<archivo_de_texto>`***

Ademas de Vi, la mayoria de las distribuciones de linux incluyen una versión mejorada del editor Vi, esta se llama Vim (Vi mejorado). Para invocar al editor Vim:

Sintaxis: **`vim`** ***`<archivo_de_texto>`***

Arranque vi preparando para un ejercicio tutorial; en todo este documento , los ejercicios usarán el archivo , ejemplo. Entonces en una terminal tipee lo siguiente:

`$vi ejemplo` <ENTER>

Limpia la ventana y muestra el contenido del archivo, ejemplo. Dado que este es un nuevo archivo, no contiene ningún texto. vi usa el caracter (~) para indicar líneas en la pantalla más allá del final el archivo.

vi usa el cursor para indicar donde su próximo comando o inserción de texto tomará efecto.
Al pie de la ventana, vi mantiene una línea de estado, llamada línea de modo. Esta línea muestra la línea corriente del archivo, el nombre del archivo, y el estado.

Modo Comando y Modo Inserción
vi tiene dos modos, modo comando y modo inserción. En el modo comando, los caracteres que ud. tipee realizan funciones tales como movimiento del cursor, cortar o copiar texto, o buscar por algún texto particular. En el modo inserción, se tipea para insertar o sobrescribir texto. Cuando arranca vi lo hace por defecto en modo comando.
Para cambiar desde modo comando a modo inserción, presione la tecla "i" (no es necesario presionar <ENTER>). vi le permite insertar texto comenzando desde la ubicación corriente del cursor. Para volver a cambiar a modo comando, presione la tecla <ESC>. Puede también usar <ESC> para cancelar un comando incompleto en el modo comando.
Desafortunadamente, vi normalmente no indica en que modo se encuentra . En el próximo ejercicio debe cambiar el indicador de modo. Si está inseguro acerca del corriente modo, puede presionar <ESC> unas cuantas veces para asegurarse que cambió el modo. Cuando vi hace sonar el beep, significa que se ha retornado al modo comando.

Ejercicio: Antes de comenzar a tipear el texto, se activa el indicador de línea de modo. Puede ser que no se requiera usarlo , pero esta le avisará si está en modo comando o en modo inserción. Tipee lo siguiente:

```
:set showmode <ENTER>
```

Nada parece haber cambiado. Cuando se encuentra en modo comando, no hay indicador, pero si entra en modo inserción , le aparecerá el modo en la esquina inferior derecha de la pantalla.

Insertando Texto
Mientras está en modo inserción, puede ingresar texto normalmente. vi reconoce unas cuantas combinaciones de teclas especiales cuando las tipea.

Backspace	Borra el carácter previo
Delete	Borra el carácter corriente

CTRL+W Borra la palabra previa
CTRL+U Borra la línea actual
<ENTER> Comienza una línea nueva

Ejercicio: Para entrar en modo inserción, presione: <i>
Note que en la esquina inferior derecha, vi indica que está en modo inserción. Entonces ingrese el siguiente texto, y recuerde presionar <ENTER> al final de cada línea. Puede usar las combinaciones de teclas especiales si comete un error.

Nuestras tiendas deterioradas fueron un pobre refugio esta noche. Cae un rayo destellante sobre la colina, fabricamos una pequeña caberna en la ladera de la montaña.

Después de ingresar este texto, presione <ESC> para volver al modo comando. Note que la linea de modo se limpia.

Movimiento del Cursor
Ud. necesitará mover el cursor en todo el archivo. Puede mover el cursor en modo comando. vi tiene algunos comandos diferentes para mover el cursor. Las cuatro teclas básicas aparecen mas adelante. Puede también usar la tecla backspace y barra espaciadora para mover a izquierda y derecha, respectivamente.

k	mueve una línea arriba
h	mueve un carácter a la izquierda sobre la linea
l	mueve un carácter a la derecha sobre la linea
j	mueve una línea abajo

Borrando Texto
En algunos casos es necesario borrar el texto editado. Para hacer esto, primero mueva el cursor tal que este cubra el primer carácter del grupo que desea borrar, entonces tipee el comando deseado de la tabla siguiente.

Algunos comandos de Borrado

x	borra solo el carácter actual.
D	borra hasta el final de la linea.
dd	borra la linea corriente.

Deshaciendo
Ocasionalmente en forma accidental se puede insertar un comando o borrar algún texto y desea restaurar lo que habia anteriormente o volver atrás el comando. vi le permite deshacer el ultimo cambio de texto o deshacer el ultimo comando, lo cual se ejecuta tipeando u. Note, no obstante, que vi solo recuperará el último cambio del texto.

Grabando su Trabajo
vi tiene varias maneras para grabar los cambios. Además de guardar su trabajo antes de salir, también es buena idea guardar su trabajo periodicamente. Una falla en la energía electrica o una caída del sistema puede causar que pierda su trabajo. Desde modo comando, tipee :

`:w` <ENTER>

graba su trabajo ("w" es por "write"). Similarmente, para salir de vi use el comando:

`:q` <ENTER>

Una forma más corta y directa para el mencionado comando es "ZZ" (SHIFT ZZ).
En algunos casos ud.desea abandonar los cambios hechos desde la última vez que grabó su trabajo (la última vez que usó el comando:w). En este caso, tipee:

`:q!` <ENTER>

lo cual le asegura que saldrá de vi sin grabar. Este comando debe ser usado con precaución, ya que abandonando vi de esta manera perderá los cambios en forma permanente.

Número de líneas
La mayoría de los comandos vi usa numero de líneas, lo cual simplifica contar el número de caracteres <ENTER> en un archivo. Se puede cortar y copiar texto por número de líneas o saltar a una cierta linea. Los números de linea pueden ser usados cuando recibe mensajes de error durante la compilación de un programa. Frecuentemente los compiladores imprimen el número de línea, de este modo se puede saltar a un determinado número de línea a ver el error.

Ejercicio: Para mostrar el número de línea, ingrese el siguiente comando:

`:set number` <ENTER>

Este comando mostrará inmediatamente el número de línea en el margen izquierdo de la pantalla del editor vi.
Para quitar el numero de lineas ingrese el comando :

`:set nonumber` <ENTER>

Otros Modos de Inserción
Además del modo inserción visto anteriormente, vi emplea otros modos de inserción. Todos ellos le permiten ingresar texto; la única diferencia esta en el punto de inserción. La Tabla de abajo describe los tres modos mas comunes: append, insert, y open. Otros dos modos de inserción son el modo cambio y el modo de reemplazo. El indicador de modo muestra el modo actual.

Algunos modos de inserción

a	append	justo después del carácter actual.
i	insert	justo antes del carácter actual.
o	open	nueva línea debajo de la actual.

Buscar
Como los archivos llegan a ser largos, se necesita asistencia para localizar una instancia particular de texto, vi tiene varias características de búsqueda.

Búsqueda Simple
vi puede buscar en un archivo completo por una cadena de texto dado. vi busca hacia delante con la barra (/) o hacia atrás con el signo de interrogación (?). Ejecute la búsqueda tipeando el comando luego el string seguido de RETURN. Para cancelar la búsqueda, presione <ESC> en vez de <ENTER>.
Se puede hacer una nueva búsqueda tipeando n (hacia delante) o N (hacia atrás). También, cuando vi alcanza el final del texto, continua buscando desde el comienzo. Esta característica es llamada "wrapscan".

Por una explicación más detallada de los comandos del editor Vi, consultar el manual de Vi disponible en la pagina web del curso.

Otros editores de línea de comando
Existen multitud de editores de línea de comandos para Unix/Linux que podríamos utilizar. El elegir uno en particular dependerá de gustos personales y disponibilidad de los mismos en el sistema que estemos usando.

En particular cabe destacar un conjunto de editores que se crearon con la finalidad de resultar familiares a los usuarios del procesador de textos WordStar para DOS.
Estos son:

- Nano.
- Pico.
- Joe.

La disponibilidad de alguno o algunos de éstos dependerá de la distribución de Linux y las opciones de instalación de la misma.

Además existen gran numero de editores orientados a brindar facilidades a los programadores, ejemplo de estos son: Emacs y jed.