Práctica 2. Redireccionamiento de Entradas y Salidas. Filtros.

1. Objetivos.

- Modificar la fuente de los datos de entrada o el destino de los resultados de las órdenes Linux.
- Filtrar el contenido de archivos de texto para seleccionar solamente parte de la información que contienen.

2. Redireccionamiento de Entradas y Salidas.

Todas las funciones de Entrada/Salida (E/S) se realizan en Linux utilizando archivos, incluidas las salidas por pantalla y las entradas desde teclado.

Por defecto, todas las entradas se hacen desde el teclado (archivo *stdin*) y las salidas se envían hacia la pantalla (archivo *stdout*). También se envían hacia la pantalla los mensajes de error (archivo *stderr*).

Los operandos **>,>>,<,<<,|** permiten obtener datos necesarios para una orden desde un archivo diferente del teclado, así como guardar los resultados en archivos en lugar de mostrarlos por pantalla.

Operando	Resultado
`orden < archivo`	*orden* utiliza como datos los contenidos en *archivo*.
`orden > archivo`	Guarda en *archivo* los resultados de *orden*. Si *archivo* ya existe, lo sobreescribe.
`orden >> archivo`	Añade los resultados de *orden* al final de *archivo*. Si no existe, lo crea.
`orden << etiqueta`	*orden* utiliza como datos los que se escriban en las líneas siguientes hasta encontrar una línea que contenga *etiqueta*.
`orden1 | orden 2`	Envía el resultado de *orden1* como dato para *orden2* utilizando una tubería o *pipe* (mediante el carácter [|]).

Ejemplos:

```
grep 'Pedro' lista_Amigos > lista_Pedros
```

Guarda en el archivo *lista_Pedros* las líneas del archivo *lista_Amigos* que contienen la palabra *Pedro*. Si *lista_Pedros* ya existía, desaparece su contenido anterior.

```
grep 'Juan' lista_Amigos | wc –l
```

Busca las líneas del archivo *lista_Amigos* que contienen la palabra *Juan* y se cuenta el número de líneas encontradas. Podría servir para contar cuantos amigos de nombre *Juan* contiene el archivo *lista_Amigos*.

3. Filtros.

Los filtros son órdenes o conjuntos de órdenes que permiten encontrar información concreta dentro de un archivo o cambiar su formato.

Expresiones regulares.

Hay órdenes que permiten buscar patrones de caracteres dentro de archivos de datos. Para indicarles qué patrones deben buscar se utilizan las llamadas *expresiones regulares*.

En las expresiones regulares pueden aparecer una serie de caracteres especiales cuyas funciones son las siguientes:

Carácter	Patrón de caracteres que representa
c	Si **c** es cualquier carácter distinto de **\[.*^]$** entonces representa una única aparición de ese carácter.
\	Elimina el significado especial del siguiente carácter.
.	Cualquier carácter.
^	Comienzo de una línea.
$	Fin de una línea.
[caracteres]	Uno cualquiera de los caracteres contenidos en la lista que aparece entre corchetes. Son patrones especiales [A-Z], [a-z] y [0-9] que representan, respectivamente, a todas las letras mayúsculas, a todas las minúsculas y a todos los caracteres numéricos. También se pueden utilizar combinaciones de estos patrones especiales.
[^caracteres]	Un carácter cualquiera que no pertenezca a la lista que aparece entre corchetes.

Ejemplos:

jpg$

Línea que finaliza con los caracteres *jpg*.

^[A-Za-z][0-9]...ZZ

Línea que comienza con una letra seguida de un número, tres caracteres cualesquiera y dos zetas mayúsculas.

[iI]magen\.doc$

Línea que contiene la cadena de caracteres *imagen.doc* o bien *Imagen.doc*. Observe que el significado especial del carácter punto (.) se elimina con la barra invertida (\ o *backslash*).

Búsqueda de patrones en archivos.

Para buscar un patrón concreto de caracteres dentro de un archivo se utiliza la instrucción grep (Global Regular Expression Pattern match).

Sintaxis: `grep expresion_regular archivo`

Por ejemplo, sea el contenido de un archivo de nombre *amigos* el siguiente:

```
juan%Burgos%18%juan@cenidet.edu.mx
Luisa%Avila%19%luisa@estudiantes.unileon.mx
Ernesto%Leon%18%ernjuan@estudiantes.unileon.mx
Luis%Burgos%20%Leon@ubu.mx
Marta%Leon%19%marLuis@estudiantes.unileon.mx
Juan%Leon%19%juan@unileon.mx
```

Observe que es una pequeña base de datos formada por un solo archivo en la que aparece un registro diferente en cada línea y en la que los campos de cada registro (nombre, localidad, edad y dirección de correo electrónico) están separados por el carácter %.

Si se desea ver la información de todos los amigos que se llaman *Juan*, podría utilizarse la orden:

```
grep '[Jj]uan' amigos
```

Los caracteres entre corchetes indican que se acepta la inicial tanto en mayúsculas como en minúsculas.

El resultado que se obtendría sería el siguiente:

```
juan%Burgos%18%juan@ubu.mx
Ernesto%Leon%18%ernjuan@estudiantes.unileon.mx
Juan%Leon%19%juan@unileon.mx
```

No solamente aparecen los amigos de nombre *Juan*, sino también *Ernesto*, que tiene los caracteres de *juan* en su dirección de correo electrónico. La solución correcta pasa por indicar a la orden grep que el patrón buscado debe estar al comienzo de la línea (que es donde se encuentra el campo nombre):

```
grep '^[Jj]uan' amigos
```

Resultado:

```
juan%Burgos%18%juan@ubu.es
Juan%Leon%19%juan@unileon.es
```

Selección de campos dentro de un archivo.

La orden cut permite visualizar caracteres o campos concretos de todas las líneas de un archivo.

Sintaxis: `cut -cRango archivo`

`cut [-dcaracter] –fRANGO archivo`

Modificadores:

`-cRANGO`	Selecciona los caracteres dentro del RANGO.
`-dcarácter`	Especifica el separador de campos.
`-fRANGO`	Selecciona los campos dentro del RANGO.

RANGO puede tener los siguientes formatos:

número-	Todos los caracteres o campos desde *número* hasta el último.
número1-número2	Todos los caracteres o campos desde *número1* hasta *número2*.
número1,número2, ...	Solamente los campos *número1*, *número2*, ...

El primer campo o carácter corresponde al número 1

Para una aplicación que envíe correos electrónicos a toda la lista de amigos, podría ser interesante obtener las direcciones de todos ellos a partir del archivo *amigos*:

```
cut –d% -f4 amigos
```

El resultado es:

juan@ubu.es
luisa@estudiantes.unileon.es
ernjuan@estudiantes.unileon.es
Leon@ubu.es
marLuis@estudiantes.unileon.es
juan@unileon.es

Si, además, se deseara tener el nombre de cada amigo, la orden adecuada sería:

```
cut –d% -f1,4 amigos
```

Cuyo resultado es:

juan%juan@ubu.es
Luisa%luisa@estudiantes.unileon.es
Ernesto%ernjuan@estudiantes.unileon.es
Luis%Leon@ubu.es
Marta%marLuis@estudiantes.unileon.es
Juan%juan@unileon.es

Ordenamiento del contenido de un archivo.

Se pueden ordenar alfabéticamente las líneas de un archivo utilizando la orden sort.

Sintaxis: `sort [-ru] archivo`

Modificadores:

- `-r` Orden descendente
- `-u` Elimina las filas duplicadas
- `-ru` Orden descendente eliminando las filas duplicadas

El ordenamiento se realiza en base a los valores numéricos de los caracteres ASCII, por ello, el carácter **Z** (mayúscula) irá antes del carácter **a** (minúscula) en el archivo ordenado de forma ascendente.

Traducción de caracteres.

La orden tr facilita la tarea de reemplazar unos caracteres por otros dentro de un archivo.

Sintaxis: `tr 'conjunto inicial' 'conjunto final' < archivo`

`tr –d 'conjunto inicial' < archivo`

Cuando se utiliza sin ningún modificador, tr lee el archivo *archivo* y reemplaza los caracteres contenidos en el conjunto inicial por los caracteres del conjunto final. Se visualiza el resultado pero no se modifica el archivo original.

Con el modificador **–d**, lee el archivo *archivo* y elimina los caracteres contenidos en el conjunto inicial. Se visualiza el resultado, pero no se modifica el archivo original.

Ejemplos:

```
tr '%' '&' < amigos >amigos_2
```

Cambia el carácter **%** (separador de campos) del archivo *amigos* por el carácter **&** y guarda el resultado en el archivo *amigos_2*. El contenido de *amigos* no varía, mientras que el de *amigos_2* se puede visualizar con la orden cat y es el siguiente:

```
juan&Burgos&18&juan@ubu.es
Luisa&Avila&19&luisa@estudiantes.unileon.es
Ernesto&Leon&18&ernjuan@estudiantes.unileon.es
Luis&Burgos&20&Leon@ubu.es
Marta&Leon&19&marLuis@estudiantes.unileon.es
Juan&Leon&19&juan@unileon.es
```

```
tr '[A-Z]' '[a-z]' < amigos
```

Cambia todos los caracteres en mayúsculas por caracteres en mayúsculas, visualiza el resultado, pero no guarda los cambios.

```
tr –d '@' < amigos >>amigos
```

Elimina todos los caracteres @ del archivo *amigos* y guarda el resultado añadiéndolo al final del mismo. No se visualiza el resultado por pantalla porque se redirecciona la salida (**>>**).

4. Desarrollo de la práctica.

Obtención del número de caracteres, palabras y líneas de un archivo.

1. Utiliza la ayuda para ver qué archivos *man* contienen la palabra *file*.
2. En lugar de ver la salida en pantalla, envíala a un archivo que se llame *lista.file*.
3. Obtén el número de archivos de ayuda que hacen referencia a la palabra *file*. (Para ello cuenta el número de líneas del archivo *lista.file*). Utiliza la orden wc, cuya sintaxis es: `wc [-lwm] archivo`

   Modificadores.

`-l`	Devuelve el número de lineas de *archivo*.
`-w`	Devuelve el número de palabras de *archivo*.
`-m`	Devuelve el número de caracteres de *archivo*.
Sin modificadores	Devuelve los tres valores anteriores.

4. Utiliza una tubería (carácter `[|]`) para ver el mismo resultado sin necesidad de usar el archivo intermedio *lista.file*. El resultado de `man –k file` se debe enviar a la orden `wc`.

Búsqueda de patrones.

1. Visualiza las líneas de *lista.file* que contienen la palabra *password*:
2. Visualiza las líneas que comienzan con la letra *o*:
3. Visualiza las líneas que terminan con letra mayúscula:
4. Cuenta las líneas de *lista.file* que contienen la palabra *password*. Obtén el resultado empleando una sola línea de órdenes y sin utilizar el archivo *lista.file*.

Selección de campos y ordenamiento.

1. Visualiza los caracteres de las columnas 1 a la 2, 27, 44 y de la 54 a la 55 del archivo *cortaletras* que se encuentra en la carpeta temporal y que contiene el siguiente texto:

```
PARA RAQUEL ESTABA MUY CLARO EL HECHO DE QUE AUGUSTO CESAR
QUERIA CASARSE CON SU MADRE, LUISA MARY.
ESPABILATE, LE AZUZABA SU TIA LINA. SI VAS A
MURCIA ANTES QUE LUISA MARY, HALLARAS UN
BIZCONDE QUE TE AYUDARA A ENCONTRAR SOLUCION A TU MAL.
>;-----------------------PEDRO DE LA BARCA
```

2. Visualiza los nombres de las órdenes referenciadas en el archivo *lista.file*. Observa que el carácter separador de campos es **–**(guión) y que las órdenes aparecen en el primer campo de cada línea.

3. Repite el ejercicio anterior pero de manera que en lugar de visualizar el resultado, éste se almacene en el archivo *lista.ordenes*. Si ese archivo no existe, debe crearlo. Si ya existe, debe sobreescribirlo.

4. Escribe la línea de órdenes necesaria para cortar del resultado de `man –k file` el nombre de las instrucciones y presentarlas en pantalla ordenadas alfabéticamente.

Traducción de caracteres

1. Utiliza la orden apropos para ver los archivos de ayuda que hay con la palabra *password*. Guarda el resultado en un archivo de nombre *lista*.

2. Añade al final de *lista* el resultado de la misma operación con la palabra *passwd*.

3. Ordena el archivo *lista* y elimina las filas duplicadas. Guarda el resultado en un archivo de nombre *lista.sinduplicar*.

4. Cambia todos los caracteres **–** (guión) que aparezcan en *lista.sinduplicar* por caracteres @ y guarda el resultado en *lista2*.

5. Cambia las letras minúsculas de *lista2* por mayúsculas y deja el resultado en *lista3*.

6. Borra los espacios en blanco de *lista3*. Deja el resultado en *lista2*.

7. Vuelca en pantalla el contenido de *lista2*.

Obtención de una lista de alumnos.

1. Copia el archivo */etc/passwd* a tu directorio.

2. Busca todos los identificadores de usuario (*userid*) que comienzan con dos letras, terminan con dos números y tienen cinco y solamente cinco caracteres (son los que

corresponden a cuentas de alumnos). Cuenta el número de alumnos que aparecen. Utiliza una sola línea de órdenes y ningún archivo intermedio.

3. Selecciona la columna que contiene el *userid* de cuentas de alumnos y la que tiene la ruta del directorio de usuario. Ordena esas dos columnas de forma alfabética eliminando las posibles filas duplicadas. Guarda el resultado en *lista.clase*. No utilices ningún archivo intermedio.