Práctica PR1006

En esta práctica debes indicar el comando que introducirías en Bash para realizar lo que se pide en cada uno de los puntos. Debes introducir la respuesta en el espacio reservado para código después de cada punto.

Para descargar este archivo en formato `md` hazlo desde la url `https://vgonzalez165.github.io/apuntes_iso/UT10_Linux_Instalacion/15_pr1006.md`. Recuerda que puedes utilizar el comando `wget` de Bash para descargar ficheros de Internet.

Ejercicio 1. Obtención de correos electrónicos de un fichero

En esta página web podrás descargar un fichero con datos de usuarios entre los cuales se incluye el correo electrónico.

Si lo abres verás que es un fichero CSV con tres campos: correo electrónico, nombre y apellido. Estos campos están entre comillas y separados por comas.

```
"michael_warder@msn.com", "", ""
"jgranos@hotmail.com", "John", "Granos"
"spakzilla@cableone.net", "john", "skr"
"lowkell@gmail.com", "Lowell", "Feld"
```

Intenta obtener los siguientes ficheros:

- Un fichero que únicamente incluya los correos electrónicos.

```
sed -E '
s/^"//
s/".*$//' articulocorreos.csv > correosfichero
```

- Un fichero que incluya solo los correos electrónicos de los usuarios cuyo nombre no está en blanco.

```
cat articulocorreos.csv | sed 's/^"//;/"."".*$/d;s/".*$//' > correosblanco
```

- Un fichero limpio que muestre el correo electrónico y el nombre completo del usuario de la forma: `jgranos@hotmail.com – John Granos`

```
cat articulocorreos.csv | sed -E 's/^"//;s/"."(.*)"."(.*)"/ - \1 \2/'
```

Consejos:

- No olvides eliminar la primera línea que tiene el encabezado

- Este ejercicio es muy sencillo porque sabemos que todas las líneas incluyen exactamente 6 caracteres de comillas por lo que puedes utilizarlas como referencia en una expresión regular. El mail está entre la primera y la segunda, el nombre entre la tercera y la cuarta y el apellido entre la quinta y la sexta.
- Puedes utilizar el **backreferencing** para seleccionar qué quieres que se muestre, es decir, toda la línea se ajusta a la expresión regular y con *backreferencing* la sustituyes con la parte que te quieras quedar.

Ejercicio 2. Generación de un fichero JSON

Un formato de datos muy popular en los últimos años es **JSON**. En él se exponen los datos como un par clave-valor. Su popularidad se debe a que es un formato claro y legible, tiene menos sobrecarga que XML, se procesa rápidamente y la mayoría de los lenguajes de programación disponen de múltiples herramientas para trabajar con este tipo de fichero.

El objetivo es extraer la información del fichero anterior y volcarla en un fichero JSON. No vamos a entrar en detalles del formato de JSON (si te interesa en la web de Mozilla se explica la sintaxis de este lenguaje), pero el fichero que generemos tiene que ser de la siguiente forma:

```
{
"Correos": [
    {"mail": "jgranos@hotmail.com", "nombre": "John Granos"},
    {"mail": "sparkzilla@cableone.net", "nombre": "john skr"},
    …
]}
```

```
echo '{"Correos": [' > ficheroson.json
cat articulocorreos.csv | sed -E 's/^"/\{"mail"\: "/;s/"."(.*)"."(.*)"/",
"nombre"\: "\1 \2"},/' >> ficheroson.json
echo ']}' >> ficheroson.json
```

Ejercicio 3. Jugando con palabras

En Linux hay un diccionario incluido que se encuentra en el fichero `/usr/share/dict/words` y en español en `/usr/share/dict/spanish` (si no tuvieras este último lo puedes instalar con el paquete `wspanish`).

Extrae las siguientes palabras del diccionario:

- Comenzamos con la fácil, busca todas las palabras de cuatro caracteres que están formadas por una misma sílaba repetida. Por ejemplo, *papa* o *nene*.

```
cat /usr/share/dict/spanish | grep -E '^(..)\1$'
```

- Busca todos los **palíndromos** de 6 caracteres. Recuerda que un palíndromo es una palabra que se lee igual de izquierda a derecha que de derecha a izquierda.

```
cat /usr/share/dict/spanish | grep -E '^(.)(.)(.)\3\2\1$'
```

- Busca todos los palíndromos de 5 caracteres

```
cat /usr/share/dict/spanish | grep -E '^(.)(.).\2\1$'
```

- Averigua el número de palabras que tienen 3 consonantes consecutivas

```
cat /usr/share/dict/spanish | grep -c "[bcdfghjklmnñpqrstvwxyz]
[bcdfghjklmnñpqrstvwxyz][bcdfghjklmnñpqrstvwxyz]"

cat /usr/share/dict/spanish | grep -Ec "[bcdfghjklmnñpqrstvwxyz]{3}"
```

- Obtén una estadística con el número de palabras del documento en función de su longitud, es decir, averigua cuantas palabras tiene de 1 letra, de 2 letras, de 3 letras, ...

```

```

cat /usr/share/dict/spanish | grep -c '^.$' cat /usr/share/dict/spanish | grep -c '^..$' cat /usr/share/dict/spanish | grep -c '^...$' cat /usr/share/dict/spanish | grep -c '^....$' cat /usr/share/dict/spanish | grep -c '^.....$'

```
## Ejercicio 4. Exportar información del sistema a JSON

El directorio `/proc` es un directorio especial de Linux cuyos ficheros son
especiales y representan información del sistema. Es decir, no son ficheros
reales, sino que cuando el usuario accede a su contenido en realidad está haciendo
una consulta al kernel para obtener algún tipo de información concreta del
sistema.

Por ejemplo, el fichero `/proc/meminfo` nos mostrará información sobre la memoria
del sistema.
```

victor@ubuntu:~$ cat /proc/meminfo MemTotal: 26211940 kB MemFree: 26014000 kB MemAvailable: 25837472 kB Buffers: 6652 kB Cached: 66384 kB SwapCached: 0 kB Active: 21072 kB

```
El objetivo es utilizar el comando `sed` para extraer la información de este
fichero y generarla en **formato JSON**. El fichero JSON deberá tener una
estructura similar a la siguiente:
```

```json
{
    "MemTotal": "3064652 kB",
    "MemFree": "1795144 kB",
    …
}
```

```
echo '{' > ficherocinco.json
cat /proc/meminfo | sed -E 's/^(.*):([[:blank:]]*)(.*)$/"\1":\2"\3",/'
echo '}' >> ficherocinco.json
```

Ejercicio 5. Estadísticas de un fichero

En la esta página web puedes obtener el Quijote completo en formato texto. Para descargar un archivo en Linux recuerda que debes utilizar el comando `wget`.

Vamos a utilizar expresiones regulares para extraer algunas estadísticas de este fichero. Para algunos ejercicios necesitarás combinar varios comandos mediante tuberías.

Necesitarás el comando `wc`, que sirve para contar, y también te puede ser útil el comando `uniq`, que, tomando un fichero de entrada, elimina todas las líneas repetidas.

Para algunas cuestiones también necesitarás convertir el texto en un listado de palabras (una por línea). Para conseguir esto deberás eliminar todos los símbolos de puntuación y también reemplazar los espacios por saltos de línea. El salto de línea se representa por el carácter `\n`.

Las tareas que tienes que realizar son:

- Averigua cuántas palabras diferentes tiene el Quijote.

  ```
  cat el_quijote.txt | sed 's/ /\n/g; s/,//g; s/;//g; s/://g; s/.//g; s/!//g'|
  sort | uniq | wc -l

  cat el_quijote.txt | sed 's/ /\n/g; s/[,.:;!]//g'
  ```