Sintaxis y Semántica de los Lenguajes

Trabajo práctico N*2

Nombre: Ramiro Nicolas

Apellido: Remersaro

Legajo: 214.055-0

Correo Institucional: rremersaro@frba.utn.edu.ar

Usuario GitHub: RemersaroRamiro

Link Repositorio:

https://github.com/RemersaroRamiro/-2024 K2102 2140550.git



1. Debe entregar un único script o comandos separados que resuelvan los siguientes puntos:

a. Reemplace cada punto del archivo "breve_historia.txt" por punto y salto de línea generando un nuevo archivo.

sed 's/\./\.\n/g' breve_historia.txt > breve_historia_ej1.txt

La s indica que se reemplazará los puntos (\.) por un punto y un salto de línea (\.\n) modificando el archivo original en uno nuevo.

b. Borre todas las líneas en blanco.

sed '/^\$/d' breve_historia.txt

Al no indicarse una sustitución, sed elimina las líneas que empiecen y terminen sin tener ningún carácter (la d indica borrar). Este comando también separa los números que están separados por puntos.



c. Cree un nuevo archivo: "breve_historia_2.txt" con el resultado de las operaciones a y b (redireccionamiento de la salida estándar).

sed '/^\$/d' breve historia ej1.txt > breve historia 2.txt

Simplemente edita lo producido del ejercicio 1 en un nuevo archivo, aplicando el ejercicio 2

d. Del archivo "breve_historia.txt", liste todas las oraciones que contengan la palabra "independencia" sin distinguir mayúsculas y minúsculas.

 $sed -E 's/([0-9]\{1,3\} \land [0-9]\{3\}) \land 1@/g; s \land b([0-9]\{1,2\}) \land 1@/g; s \land \land \land g' breve historia.txt \mid grep -i 'independencia'$

Primero separa en oraciones al archivo original y luego lista las líneas (ahora oraciones) que contengan independencia sin distinguir mayúsculas y minúsculas (-i). Nótese que para separar las oraciones se modifica el comando del punto 1 ya que ahora sí considera los números separados por puntos. Esto lo hace reemplazando estos puntos por un nuevo carácter, divide como el comando del punto 1 y luego vuelve a reemplazar el carácter por un punto.

```
$ sed -E 's/([0-9]{1,3}\.[0-9]{3})\./\18/g; s/\b([0-9]{1,2})\./\18/g; s/\\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sq}\synt{\sqrt{\sqrt{\sqrt{\sqrt{\sq}\synt{\synt{\sqrt{\syn{\sq\synt{\sq\ty}\synt{\syn{\synt{\syn{\synt{\synt{\synt{\synt{\synt{\s
```

e. Muestre las líneas que empiecen con "El" y terminan con "." del archivo "breve historia.txt".

grep '^El.*\.\$' breve historia.txt

Lista todas las líneas que inicien con El (^ indica inicio) y terminan con un punto (\. , \$ indica final)

```
peg MEL *\s' browe.bistoria.txt

In periodo que transcurre entre 1882 y 1880 marca un virage fundamental en la historia argentina. La acción orgánica y tenaz del poder publico durante las tres primeras presidencias constitucionales no solo puso fin a los viejos problemas que se habían debatido durante con contectado siste que inscio una era de cambio sustamistas en la estructura económica y social del pars. Esta deble faz de la acción de las minorias pobernantes correspondia a un cambio profundo de mentalidad en la generación que l'Ispó al poder después de la batalla de los constitucions de la constitución que respectado de las minorias pobernantes correspondia a un cambio profundo de mentalidad en la generación que l'Ispó al poder, esta entre de la batalla de los constitucions de la constitución, que respectado de la servicio profundo de mentalidad en la generación que l'Ispó al poder, esta entre de la batalla de los constitucións, que respectado de la constitución que respect
```

f. Sobre el mismo archivo del punto anterior, Indique en cuántas oraciones aparece la palabra "peronismo". Puede usar la opción -c para contar.

```
sed -E 's/([0-9]\{1,3\}\.[0-9]\{3\})\.\/1@/g; s/\b([0-9]\{1,2\})\.\/1@/g; s/\.\/s*/.\/n/g; s/@/\.\/g' breve\_historia.txt | grep -c 'peronismo'
```

Primero divide el archivo en oraciones y luego cuento las veces que aparece peronismo en cada línea(ahora oraciones)

g. Muestre la cantidad de oraciones que contienen la palabra "Sarmiento" y la palabra "Rosas".

```
sed -E 's/([0-9]{1,3}\.[0-9]{3})\.\/1@/g; s/\b([0-9]{1,2})\.\/1@/g; s/\.\/s*/.\/n/g; s/@/\.\/g' breve\_historia.txt | grep 'Sarmiento' | grep -c 'Rosas'
```

Primero divide el archivo en oraciones, luego encuentra las oraciones que contengan Sarmiento, y sobre eso se cuentan las que también tengan Rosas.

```
$ sed -E 's/([0-9]{1,3}\.[0-9]{3})\./\1@/g; s/\b([0-9]{1,2})\./\1@/g; s/\.\s*/.\n/g; s/@/\./g' breve_historia.txt | grep 'Sarmiento' | grep -c 'Rosas' 1
```

h. Muestre las oraciones que tengan fechas referidas al siglo XIX.

```
 sed -E 's/([0-9]\{1,3\}\.[0-9]\{3\})\.(1@/g; s/\b([0-9]\{1,2\})\.(1@/g; s/\.(s*/.\n/g; s/@/\./g') breve historia.txt | grep -E '\b18[0-9]\{2\}\b'
```

Primero divide el texto en oraciones y luego muestra las que coincidan con algún año del 1800 a 1899

```
Lis principo periodico de se públicaron en Bumbo Aires, El Telegrafo Nercentil (2001) y el Semanario de Apricaliuma, Cobarcio (2002), reflegiban has nuevas prescupaciones por los problemas económicos y sociales. Designado virrey o 1804, el amegia de Sobremont Edución effertar la nuevas inde en agento to appear de puso al virrey y designó una Junta de Gobierno.

In 1813 er remito la Asabiba General Constituyente, que tunó muneroas decisiones que repetiban el predestinio de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tunó muneroas decisiones que repetiban el predestinio de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tunó muneroas decisiones que repetiban el predestinio de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tunó muneroas decisiones que repetiban el predestinio de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tunó muneroas decisiones que repetiban el predestinio de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tunó de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tuno de la constitución de los propessitas.

In 1813 er remito la Asabiba General Constituyente, que tental propessita de la constitución de los propessitas.

In 1813 er remito la Constitución que los propessitas.

In 1814 esta de la constitución que la constitución de los propessitas.

In 1815 esta despeta que la constitución de los propessitas.

In 1815 esta despeta que la constitución de los propessitas.

In 1815 esta despeta que la constitución de los propessitas.

In 1815 esta despeta que la constitución de los propessitas.

In 1815 esta despeta que la constitución propessitas.

In 1815 esta despeta que la constitución propessitas.

In 1815 esta despeta que la constitución propessitas de la constitución que los constitución que
```

i. Borre la primera palabra de cada línea. Utilice substitución con sed. La sintaxis para sustituir la primera palabra de cada línea por "nada" sería: \$sed "s/^[[a-zA-Z]]*\b//g" nombre_archivo (La "s" indica sustitución; entre los dos primeros /.../ está la expresión regular que queremos reemplazar, en este caso "/^[[a-zA-Z]]*\b"; entre el segundo y el tercer "//" se indica la expresión por la cual será reemplazada, en este caso por la palabra vacía. Finalmente la "g" indica que el cambio será en todo el archivo.

```
sed 's/^[[:alpha:]][[:alpha:]]*//' breve_historia.txt
```

El comando sigue la lógica presentada en el enunciado solo que modifique el contenido que tiene que detectar para que tome en cuenta las palabras con tilde(ya que probando en consola vi que el anterior no lo hacía). Uso [[:alpha:]] ya que equivale a [A-Za-zÁÉÍÓÚáéíóúÜüÑñ]

```
Saed 'sp\([siphas]][[siphas]][']/' breve_historia.txt

Assertions are continued as a second s
```

j. Escriba un comando que enumere todos los archivos de una carpeta que contengan extensión ".txt". (tip: pipe con el comando ls).

ls *.txt

```
$ ls *.txt
breve_historia.txt breve_historia_2.txt breve_historia_ej1.txt
```

- 2. Investigue y explique, dando ejemplos de cómo se utilizan los siguientes elementos en bash:
- Variables.

Las variables en Bash se utilizan para almacenar valores. Se declaran sin ningún tipo de tipo de datos y se asignan valores con el signo de igualdad (=). Por ejemplo: nombre="Juan"

- Sentencias condicionales.

Las sentencias condicionales se utilizan para tomar decisiones basadas en condiciones. Por ejemplo:

```
if [ condicion ]; then
código si la condición es verdadera
elif [ otra_condicion ]; then
código si la otra condición es verdadera
else
código si ninguna de las condiciones anteriores es verdadera
fi
```

- Sentencias cíclicas.

Las sentencias cíclicas se utilizan para repetir un bloque de código mientras se cumpla una condición. Por ejemplo:

```
while [ condicion ]; do código a ejecutar mientras la condición sea verdadera done
```

- Subprogramas

```
Los subprogramas en Bash se definen utilizando funciones. Pueden aceptar parámetros y devolver valores. La sintaxis básica es:
nombre_funcion() {
    código de la función
}
nombre_funcion() -> llamado a la funcion
```