Estudiante: Ignacio Scarfo

UTN – FACULTAD REGIONAL BUENOS AIRES
Materia: Sintaxis y Semántica de los Lenguajes– 2024
CURSO K2102

TRABAJO PRÁCTICO N°2

Expresiones regulares y expresiones regulares extendidas en Bash

---

Scarfo, Ignacio Alejo 214.123-1 iscarfo@frba.utn.edu.ar

FECHA DE PRESENTACIÓN: 25/06/2024 FIRMA PROFESOR

FECHA DE DEVOLUCIÓN: CALIFICACIÓN

Usuario de GitHub: iscarfo

Link al Repo: https://github.com/iscarfo/2024_K2102_2141231

Estudiante: Ignacio Scarfo

Punto 1:

A- El comando sed busca cada punto en el archivo breve_historia.txt y lo reemplaza por un punto seguido de un salto de línea en el archivo breve_historia_mod.txt que es el nuevo archivo creado, luego de ser modificado. El flag g asegura que se realice la sustitución en todas las ocurrencias de puntos en cada línea del archivo.

B- Para este punto pensé en un comando de tipo: "sed -i '/^ $/d' breve_historia.txt", ya que sed elimina todas las líneas en blanco del archivo breve_historia.txt sin crear un nuevo archivo. Sino otra forma de realizarlo podría ser creando un nuevo archivo modificado, pero como no lo dice la consigna me quedé con esta opción.

C- En este comando busque combinar los dos anteriores utilizando la barra vertical y aplicándolos para crear un nuevo archivo modificado.

D- En este punto se empieza a utilizar el comando grep para listar la palabra independencia, en donde se usa el "-i" al comienzo para que ignore si se trata de mayúsculas o minúsculas, el –o que muestra solo la parte de la línea que coincide con el patrón y el –n que añade el número de línea al principio de cada coincidencia.

E- En este caso primero elimino los caracteres CR y luego listo las líneas que empiecen con El y terminen con un punto. En este caso no me aparece ningún match en el archivo fuente

F- En este caso se realiza algo parecido al caso anterior utilizando el flag –c para contar la cantidad de veces que aparezca la palabra peronismo y el –i para que ignore mayúsculas y minúsculas.

G- En este comando busco un caso muy parecido al anterior, teniendo en cuenta que la palabra Rosas y Sarmiento estén en la misma oración ignorando mayúsculas y minúsculas

H- En este comando lo que busque es marcar en el texto todas las veces que aparece un año del siglo XIX en el texto utilizando el símbolo de potencia de expresiones Reg. extendidas

I- En este comando se elimina toda primera de cada oración, cabe aclarar que se utiliza la g para que se aplique a todo el archivo

J- En este comando con el pipe se logra enlistar todos los archivos de texto de una misma carpeta

Punto 2:

Variables:

En Bash, las variables son contenedores que almacenan datos. Pueden contener cadenas de texto, números u otros tipos de datos. Algunas características importantes de las variables en Bash son:

- Flexibilidad: No necesitas declarar el tipo de una variable antes de usarla. Puedes asignar cualquier tipo de dato a una variable y cambiar su tipo en cualquier momento.
- Convención de nombres: Las variables en Bash suelen seguir una convención de nombres en minúsculas, utilizando guiones bajos para separar palabras en nombres de variables compuestas.
- Ámbito de las variables: Por defecto, todas las variables en Bash son globales, pero puedes limitar su ámbito a una función utilizando la palabra clave local.

Estudiante: Ignacio Scarfo

Sentencias condicionales:

Las sentencias condicionales en Bash, como if-else, te permiten ejecutar diferentes bloques de código dependiendo de si se cumple una condición o no. Algunas características importantes de las sentencias condicionales en Bash son:

- Evaluación de expresiones: Las condiciones en las sentencias condicionales de Bash se evalúan utilizando comandos de prueba como [o [[, que pueden evaluar expresiones booleanas.
- Operadores de comparación: Puedes utilizar una variedad de operadores de comparación, como -eq, -ne, -lt, -gt, -le, -ge, para comparar números, y =, ==, !=, para comparar cadenas.
- Anidamiento: Puedes anidar sentencias condicionales dentro de otras, lo que te permite construir lógica condicional más compleja.

Sentencias cíclicas:

Las sentencias cíclicas en Bash, como for y while, te permiten repetir bloques de código un número determinado de veces o hasta que se cumpla una condición. Algunas características importantes de las sentencias cíclicas en Bash son:

- Iteración controlada por secuencia: El bucle for itera sobre una secuencia de elementos, como números, elementos de una lista o archivos en un directorio.
- Iteración basada en condición: El bucle while itera mientras se cumple una condición específica.
- Modificación del contador: Dentro de un bucle, puedes modificar el contador o la condición de iteración para controlar el número de repeticiones.

Subprogramas (Funciones):

Las funciones en Bash te permiten organizar y reutilizar tu código. Algunas características importantes de las funciones en Bash son:

- Declaración de funciones: Puedes declarar una función utilizando la sintaxis nombre_funcion() { ... }.
- Argumentos de función: Puedes pasar argumentos a una función cuando la llamas, y dentro de la función puedes acceder a estos argumentos utilizando $1, $2, etc.
- Valores de retorno: Puedes devolver un valor desde una función utilizando el comando return, y luego acceder a este valor fuera de la función utilizando $?.
- Ámbito de las funciones: Las variables declaradas dentro de una función por defecto son locales a esa función, a menos que se declaren como global.

*Ejemplos de los 4 tópicos en los casos de uso