

Obligatorio 1. Taller de tecnologías 1. Universidad ORT

21/05/2024

DALLA RIZZA (230723), TERZAGHI (343233) , SAN ANDREA (258053)

Este documento describe la estructura, el funcionamiento del script de Bash. El script permite listar y crear usuarios, configurar letras para consultas específicas, y ejecutar diferentes algoritmos. Incluimos una sección donde detallamos el funcionamiento de algunos de los comandos elegidos para la tarea.

Estructura del Script:

- **Creación de un Archivo con Usuarios y Contraseñas (Lineas 4 a 7):**
 - Se crea un archivo users.txt que contiene una lista de usuarios y contraseñas predefinidas. Se utiliza el marcador EOF junto con CAT para redirigir el string a un archivo users.txt (en caso de no existir, el operador > lo crea y añade el texto). Esto es necesario para poder acceder a la app por primera vez.
- **Variables Globales (Lineas 9 a 13):**
 - Estas variables se utilizan para almacenar configuraciones durante la ejecución del script. (necesario para la implementación de varias de las características)
- **Menú de Inicio (Lineas 32 a 91):**
 - Se muestra un menú interactivo que permite al usuario elegir entre varias opciones, como listar usuarios, crear un nuevo usuario, configurar letras específicas, consultar un diccionario, entre otros. Utilizamos la estructura de control CASE para determinar que función llamar dependiendo del input del usuario.
- **Funciones (Lineas 93 a 303):**
 - Cada funcionalidad se encapsuló en funciones separadas para mejorar la claridad y la mantenibilidad del código.
- **Punto de entrada al programa (Lineas 305 a 309):**
 - Se ejecuta la función logueo para solicitar el usuario y contraseña, en caso de que el retorno se evalúe como verdadero, se llama a la función que define el menú.

Detalles de Funciones Específicas:

- **ingresarVocal:**
Permite al usuario ingresar una vocal y la almacena en la variable global VOCAL.
- **listarPalabrasConVocal:**
Filtra y lista palabras que contienen solo la vocal especificada. También calcula y muestra el porcentaje de palabras que cumplen con esta condición respecto al total de palabras en el diccionario.
- **algoritmo1:**
Solicita una cantidad de datos, los ingresa, y luego calcula el promedio, el menor y el mayor de los datos ingresados. Usa operaciones aritméticas básicas y bc para manejar la precisión decimal.
- **algoritmo2:**
Solicita una palabra, la convierte a minúsculas, la invierte y verifica si es un palíndromo comparando la palabra original con la invertida.
- **configurarLetraInicio, configurarLetraFinal, configurarLetraContenida:**
Permiten configurar letras específicas para realizar búsquedas en el diccionario. Cada función solicita una letra al usuario y la almacena en la variable correspondiente.
- **consultarDiccionario:**
Consulta y filtra palabras del diccionario basado en las letras configuradas. Calcula y muestra el porcentaje de palabras que cumplen con las condiciones especificadas.
- **listarUsuarios:**
Lista todos los usuarios almacenados en el archivo users.txt usando cut para extraer solo los nombres de usuario.
- **crearUsuario:**
Permite crear un nuevo usuario ingresando un nombre de usuario y una contraseña. Verifica que el usuario no exista previamente y que los campos no estén vacíos antes de agregar el nuevo usuario al archivo.
- **logueo:**
Maneja la autenticación del usuario solicitando nombre de usuario y contraseña, verificando estos datos contra el archivo users.txt.

Lista de Comandos destacados que utilizamos:

- **cat > users.txt << EOF ... EOF:**
Utiliza el comando cat en combinación con un marcador de fin (EOF) para crear y escribir directamente en el archivo users.txt. Todo el texto entre << EOF y EOF se guarda en el archivo (Se pasa como argumento a CAT).
- **cat diccionario.txt | grep "^[^aeiou]*\$VOCAL[^aeiou]*\$":**
Usa cat para leer el archivo diccionario.txt y grep para filtrar líneas que contienen solo la vocal especificada, evitando otras vocales. Se destaca el uso del pipe (|) que permite que el segundo comando se ejecute sobre el resultado del primero y regex para evaluar cada palabra
- **wc -l:**
WC (Word Count) permite contar las líneas, palabras y caracteres de un archivo. El -L permite especificar que solo nos devuelva la cantidad de líneas.
- **"scale=3; \$cumplen / \$total * 100" | bc:**
Calcula el porcentaje de palabras que cumplen con las condiciones usando bc, una calculadora de precisión arbitraria. Se establece la precisión decimal con scale. BC es bastante dinámico, dentro de la cadena de texto se puede incluso declarar variables, asignarle valores y ejecutar operaciones sobre ellas.
- **tr '[:upper:]' '[:lower:]':**
Usa tr (translate) para convertir todos los caracteres de palabra a minúsculas. TR tiene dos argumentos, el primero es el carácter a sustituir y el segundo el sustituto. Hay terminología especial como la usada en este caso para referirse a letras mayúsculas y minúsculas.
- **rev:**
Sirve para invertir una cadena de caracteres.
- **cut -d: -f1:**
Permite cortar una palabra, con el -d se especifica el punto de corte y el -f1 especifica que "mitad" se va a seleccionar