



Mario Enrique Montes Jacuinde

Seminario de solución de problemas de sistemas operativos

Reporte practica 1

Javier Rosales Martínez

D06

Antecedentes:

Se debía de implementar un código que hiciera que un archivo de texto con un formato especificado se convirtiera en otro archivo txt diferente con otro formato eliminado y modificando ciertas partes de ese mismo archivo.

Metodologías:

Librerías utilizadas: se usó la librería `os` para que se pudiera verificar si el archivo de entrada existe en la ruta especificada antes de que fuera procesado.

Funciones utilizadas: la función `hex_dec(hex_str)`: La función lo que hace es que convierte una cadena en formato hexadecimal a formato decimal.

`dato_hex(dato_str)`: esta función convierte los últimos datos a hexadecimal.

`Procesa_linea(línea)`: esta función procesa cada línea del archivo de entrada, extrae los números hexadecimales de la primera parte de las líneas del archivo y los elimina. También extrae la segunda parte la cual corresponde al nombre, devuelve la cadena formada de la forma que se debía de implementar.

Algoritmos: Se utilizó un algoritmo para la extracción de datos, lo utiliza usando comas para separar cada elemento

Se uso otro para convertir los números hexadecimales a números decimales y también para que los últimos datos se conviertan a hexadecimales.

Diseño: para la ejecución del programa se diseño para que se ejecutara línea por línea, con esto se aseguraba que cada formato se convirtiera a la forma que debía de la mejor manera.

También se uso una verificación para que se pudiera notificar a un usuario en caso que el archivo de entrada no exista.

Conclusión:

El código ya procesa el archivo txt, lo lee, y lo modifica creando uno nuevo con las especificaciones que se pidieron modificando en hexadecimal, con nombres y con números decimales.

Unas posibles mejoras pueden ser que sea una optimización del rendimiento, que si el archivo es mas grande puede tardarse mas en hacer todas las modificaciones, también se pudiera hacer que el archivo resistiera en dado caso de agregar alguna otra línea con caracteres diferentes no especificados los pueda soportar sin problemas.

Código:

```
import os

# Esto convierte un el número hexadecimal a decimal
def hex_dec(hex_str):
    return int(hex_str, 16)

# convierte decimal a hexadecimal
def dato_hex(dato_str):
    # Convertir cada dato en su equivalente hexadecimal
    return ''.join([format(int(dat), 'X') for dat in dato_str.split('.')])

# procesa línea del archivo de entrada
def procesa_linea(linea):
    partes = linea.strip().split(',')

    # Obtiene la primera parte de números hexadecimales
    hex_numeros = partes[0].split(':')

    # Elimina cualquier caracter que no sea parte del número hexadecimal
    hex_numeros = [num.split('/')[0] for num in hex_numeros]

    # Extrae la segunda cadena (nombre o texto)
    segundo_string = partes[2].strip()

    # Extrae los ultimos datos
    ulti_datos = partes[-1].strip()

    # Convierte los números hexadecimales a decimales
    dec_numeros = [hex_dec(num) for num in hex_numeros]

    # Convierte los ultimos datos a hexadecimal
    hex_ulti_datos = dato_hex(ulti_datos)

    # Devuelve el formato final: segunda cadena, números en decimal y ultimos datos en hexadecimal
    return f"{segundo_string} : {' : '.join(map(str, dec_numeros))} : {hex_ulti_datos}"
```

```
# Rutas de los archivos de entrada y salida
input_file = 'C:/Users/Usuario/Desktop/Universidad/S.S.O/prueba2.txt'
output_file = 'C:/Users/Usuario/Desktop/Universidad/S.S.O/resultado.txt'

# Verifica si el archivo de entrada existe
if os.path.exists(input_file):
    # Abre el archivo de entrada para lectura y el archivo de salida para escritura
    with open(input_file, 'r') as infile, open(output_file, 'w') as outfile:
        # Lee y procesa cada línea del archivo de entrada
        for line in infile:
            processed_line = procesa_linea(line)
            # Escribe la línea procesada en el archivo de salida
            outfile.write(processed_line + '\n')
        # Indica que el procesamiento ha sido completado
        print("Completado. Revisa el archivo de salida.")
else:
    # Muestra mensaje de error si el archivo de entrada no existe
    print(f"El archivo {input_file} no se encuentra en la ruta especificada.")
```

Referencias: Jaiswal, S. (2024, January 29). *Tutorial de conversión de tipos de datos en*

Python. <https://www.datacamp.com/es/tutorial/python-data-type-conversion>

Ismael, E. (n.d.). *Article*. <https://pywombat.com/articles/mainipucion-archivos-python>

Bravo, B. I. (2023, October 2). *Lectura y escritura de archivos en Python: Como crear, leer y escribir archivos*. freeCodeCamp.org.

<https://www.freecodecamp.org/espanol/news/lectura-y-escritura-de-archivos-en-python-como-crear-leer-y-escribir-archivos/>