

Descripción General

El proyecto consta de varios módulos que trabajan juntos para procesar un archivo de texto que contiene direcciones IPv6 e IPv4. El proceso incluye la división de líneas, la estructuración de datos, la conversión de direcciones IPv6 a decimal y direcciones IPv4 a hexadecimal, y finalmente la escritura de los resultados en un archivo de salida.

Módulo: **OpenFile.py**

Clase: **OpenFile**

- **Descripción:** Clase para abrir y cerrar archivos de manera segura usando un contexto manager (**with**).
 - **Métodos:**
 - **init(file_path, mode='r'):**
 - **Descripción:** Inicializa el contexto para abrir un archivo.
 - **enter():**
 - **Descripción:** Abre el archivo y lo retorna para su uso dentro del bloque **with**.
 - **exit(exc_type, exc_val, exc_tb):**
 - **Descripción:** Cierra el archivo al salir del bloque **with** o en caso de excepción.
-

Módulo: **File.py**

Clase: **File**

- **Descripción:** Clase para leer el contenido de un archivo.
 - **Métodos:**
 - **init(open_file):**
 - **Descripción:** Inicializa la clase leyendo el contenido del archivo.
 - **__read_file(open_file):**
 - **Descripción:** Lee el contenido del archivo y lo retorna como una lista de líneas.
-

Clase: **FileReader**

- **Descripción:** Clase para leer líneas específicas o todas las líneas de un archivo.
 - **Métodos:**
 - **read_line(line_number):**
 - **Descripción:** Lee una línea específica del archivo.
 - **read_all_lines():**
 - **Descripción:** Lee todas las líneas del archivo.
-

Clase: **FileWriter**

- **Descripción:** Clase para escribir datos en un archivo.

- **Métodos:**
 - **write_file(open_file, output):**
 - **Descripción:** Escribe el contenido proporcionado en un archivo.
-

Módulo: **ProcessFile.py**

Clase: **LineSplitter**

- **Descripción:** Utilidad para dividir una línea de texto en partes basadas en un delimitador.
 - **Métodos:**
 - **split_line(line, delimiter=','):**
 - **Descripción:** Divide una línea de texto en partes usando un delimitador.
 - **Parámetros:**
 - **line (str):** La línea de texto a dividir.
 - **delimiter (str, opcional):** El delimitador a usar. Por defecto es ','.
 - **Retorna:** Una lista de subcadenas obtenidas al dividir la línea.
-

Clase: **LineStructurer**

- **Descripción:** Utilidad para estructurar datos divididos en un diccionario.
 - **Métodos:**
 - **structure_line(split_data):**
 - **Descripción:** Organiza una lista de datos divididos en un diccionario con claves predefinidas.
 - **Parámetros:**
 - **split_data (list):** Lista de datos divididos. Debe contener al menos 6 elementos.
 - **Retorna:** Un diccionario con las siguientes claves:
 - "hexadecimal_numbers": Números hexadecimales.
 - "first_string": Primera cadena.
 - "second_string": Segunda cadena.
 - "third_string": Tercera cadena.
 - "fourth_string": Cuarta cadena.
 - "decimal_numbers": Números decimales.
-

Clase: **IPv6ProcessorToDecimal**

- **Descripción:** Utilidad para procesar direcciones IPv6 y convertirlas a números decimales.
- **Métodos:**
 - **remove_mask(hexadecimal_numbers):**
 - **Descripción:** Elimina la máscara de una dirección IPv6.
 - **Parámetros:**
 - **hexadecimal_numbers (str):** Dirección IPv6 en formato hexadecimal.
 - **Retorna:** La dirección IPv6 sin la máscara.
 - **convert_hex_to_decimal(hexadecimal_numbers):**
 - **Descripción:** Convierte bloques hexadecimales de una dirección IPv6 a su representación decimal.

- **Parámetros:**
 - **hexadecimal_numbers (str):** Dirección IPv6 en formato hexadecimal.
 - **Retorna:** Una lista de enteros que representan los valores decimales de cada bloque hexadecimal.
-

Clase: **IPv4ProcessorToHexaDecimal**

- **Descripción:** Utilidad para procesar números decimales, especialmente en el contexto de direcciones IPv4, y convertirlos a hexadecimal.
 - **Métodos:**
 - **convert_decimal_to_hex(decimal_numbers):**
 - **Descripción:** Convierte octetos decimales a su representación hexadecimal.
 - **Parámetros:**
 - **decimal_numbers (str):** Cadena de números decimales separados por puntos.
 - **Retorna:** Una lista de cadenas hexadecimales en mayúsculas.
-

Clase: **FormatSingleLine**

- **Descripción:** Clase para formatear una línea de texto dividiendo, estructurando y procesando datos.
 - **Métodos:**
 - **init(splitter, structurer, ipv6_processor, ipv4_processor):**
 - **Descripción:** Inicializa la clase con instancias de **LineSplitter**, **LineStructurer**, **IPv6ProcessorToDecimal** y **IPv4ProcessorToHexaDecimal**.
 - **process_line(line):**
 - **Descripción:** Procesa una línea de texto para extraer y formatear direcciones IPv6 e IPv4.
 - **Parámetros:**
 - **line (str):** Línea de texto a procesar.
 - **Retorna:** Una cadena formateada con la segunda cadena, la representación decimal de la dirección IPv6 y la representación hexadecimal de la dirección IPv4.
-

Clase: **FormatAllFile**

- **Descripción:** Clase para procesar y formatear todas las líneas de un archivo.
 - **Métodos:**
 - **init(line_formatter):**
 - **Descripción:** Inicializa la clase con una instancia de **FormatSingleLine**.
 - **process_file(file_data):**
 - **Descripción:** Procesa todas las líneas de un archivo y retorna el resultado formateado.
 - **Parámetros:**
 - **file_data (list[str]):** Lista de cadenas que representan las líneas del archivo.
 - **Retorna:** Una cadena con el resultado formateado de todas las líneas.
-

Módulo: **main.py**

Descripción

- Este es el script principal que ejecuta el proceso completo:
 1. Abre un archivo de entrada para lectura y un archivo de salida para escritura.
 2. Lee el contenido del archivo de entrada.
 3. Procesa cada línea del archivo usando las clases **FormatSingleLine** y **FormatAllFile**.
 4. Escribe los datos formateados en el archivo de salida.
-

Flujo de Trabajo

1. **Apertura de Archivos:** Se abre el archivo de entrada para lectura y el archivo de salida para escritura.
 2. **Lectura de Datos:** Se leen todas las líneas del archivo de entrada.
 3. **Procesamiento de Líneas:** Cada línea se divide, estructura y procesa para convertir direcciones IPv6 a decimal e IPv4 a hexadecimal.
 4. **Escritura de Resultados:** Los datos formateados se escriben en el archivo de salida.
-