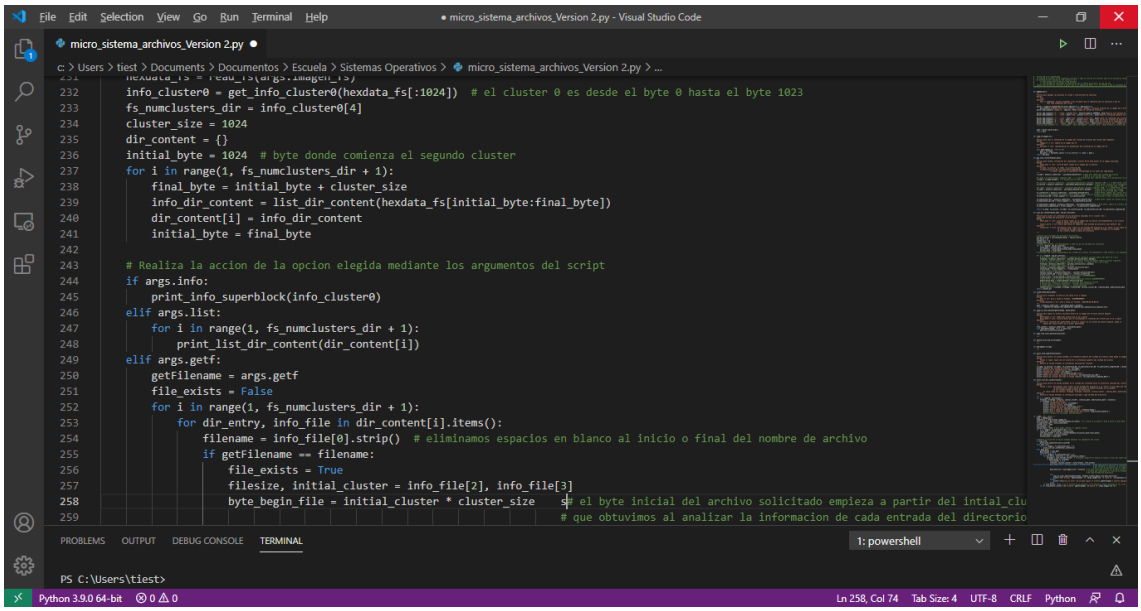
Materia: Sistemas Operativos
Profesor: Gunnar Wolf

Proyecto 3: (Micro) sistema de archivos
Alumno: Zepeda Martínez Erik



Documentación

Detalles del programa:

Uso externo de un archivo nombrado “requirements.txt” para la funcionalidad del programa.

Código fue desarrollado en:

Lenguaje Python Versión: Python 3.9.0

Visual Studio Code: 1.50.1

Introducción

Los sistemas de archivos nos dan la oportunidad de tener un control específico y ordenado sobre nuestra información. Prácticamente están presentes en cualquier tipo de sistema operativo ya que resultan una herramienta de gran importancia entre los usuarios. Ya que como es mencionado la oportunidad de modificar la información guardada es uno de los beneficios más notables.

A pesar de que es una herramienta básica en nuestro uso cotidiano debo admitir que no siempre reconocimos la importancia o la complejidad de dicho sistema. Es por eso que en este proyecto intente profundizar, entender y experimentar la lógica computacional para el uso e implementación de estas unidades de almacenamiento.

Desarrollo

A grandes rasgos la composición del programa inicia con la definición de una función que relaciona algunas letras (h, i, l, g, p y r) con las diferentes acciones que pueden realizarse en el sistema. Después se inicializan las funciones para realizar las tareas, ya sea la edición de la fecha, borrar un archivo, información general, listado de contenidos o la copia bilateral entre el sistema fs y el equipo.

1- Relación de letras y acciones.

2- Posteriormente inicializar las funciones de dicho sistema.

3- En este caso la función para leer el contenido de la imagen “fiunamfs”.

4- Seguido de una función para obtener la información del superbloque.

5- Separación de Bytes.

6- Función que muestra los contenidos del directorio.

7- Función que formatea la fecha.

8- Copia de un archivo del equipo en “fiunamfs”.

9- Copia de un archivo de la imagen “fiunamfs” al equipo.

10- Salida de la información general del sistema de archivos leído a la imagen.

11- Salida estándar en el listado del contenido hacia el directorio.

12- Impresión de mensajes sobre el manejo adecuado o inadecuado de la información.

Consideraciones

Sistema de archivos cabe en diskette de 1440 kb (1.44M)

Las cadenas de texto deben ser de los caracteres en ASCII 8-bit.

https://elcodigoascii.com.ar/codigos-ascii/numero-ocho-8-codigo-ascii-56.html

En las estructuras del disco los enteros de 32 bits se representan en little endian

Tam. de sector: 512 bytes

Tam. cluster: 2 sectores => 1024 bytes

No maneja tabla de particiones (colección de sectores consecutivos en un volumen)

Es directamente un volumen (conjunto de sectores para almacenar datos)

Maneja únicamente un directorio plano (no hay subdirectorios)

Cluster #0 es el superbloque (estructura de datos de 1024 bytes que contiene valores de configuracion)

El sistema de archivos es de asignación contigua => toda la info de los archivos está en el directorio [clusters 1 al 4]

El directorio está ubicado en los clusters 1 a 4

“=>“ cada entrada del directorio mide 64 bytes

“=>" las entradas no utilizadas se identifican con el nombre de archivo: Xx.xXx.xXx.xXx.

Después del directorio (clusters a partir del 5) es espacio de datos => => se refiere a que el contenido de cada archivo está en los bytes de los clusters a partir del 5to.