

Nombre: Marco Sebastian Solares España  
Carnet: 202004822  
[MIA]Tarea2

Primero se manda el comando a analizar a la función analyze, luego depende que tipo de comando sea se mandara a analizar de manera diferente por ejemplo:

Comando: execute

Al escribir este comando se manda a la función execute()

```
5
6 def analyze(comando):
7
8     # Analizamos el comando
9
10    token_ = re.split(" ",comando)
11
12    # Cadena en minuscula
13    token_[0] = token_[0].lower()
14
15    if token_[0] == "exec":
16        #print(token_)
17        execute(token_[1:])
18    if token_[0] == "mkdisk":
19        analyze_mkdisk(token_[1:])
20
21
22    if token_[0] == "rep":
23        rep(token_[1:])
24
25    else:
26
27        analyze(input("< "))
```

Sebastian Solares  
202004822

### Execute

Este comando funciona para abrir una archivo por medio de una ruta path que mandamos. Lee el archivo línea por línea , mostramos la información obtenida, y si tenemos algunos comandos lo ejecutamos

exec \$path>= " RUTA "

```
def execute(tk):
    tk.pop(0)
    tk.pop(0)

    #Sumamos cada valor
    path = ""
    for x in range(len(tk)):
        path = path + tk[x] + " "
    path_final = path.rstrip()

    # borramos comillas al principio y al final
    path_final = path_final[1:-1]

> with open(path_final, "r") as file: ...
    file.close()
```

Sebastian Solares  
202004822

## MKDISK

Con este comando creamos un archivo simulando un disco duro, le asignamos su tamaño, nombre, etc. Y le añadimos la información deseada.

```
def analyze_mkdisk(token_):
    """Creamos el archivo vacio"""
    with open("Salida/Hard_disk.dsk", "wb") as file:
        size_kilobyte = 1024
        for i in range(0, 5000):
            file.write(b'\x00' * size_kilobyte)
        file.close()

    """Creamos el MBR"""
    with open("Salida/Hard_disk.dsk", "rb+") as file:
        file.seek(0, 0)
        size_file = os.path.getsize("Salida/Hard_disk.dsk")
        print(size_file)
        date_file = time.ctime(os.path.getctime("Salida/Hard_disk.dsk"))
        num_random = random.randint(0, 9)

        data_pack = struct.pack("i 24s 8s", size_file, date_file.encode(), str(num_random).encode('utf-8'))
        file.write(data_pack)
```

Sebastian Solares  
2020004822

Rep

Este comando nos permite buscar un archivo y leer línea por línea dicho archivo, como por ejemplo queremos leer la información de un disco duro creado con mkdisk.

```
def rep(token_):  
    #Vamos a abrir y leer un archivo binario  
    with open("Salida/Hard_disk.dsk","rb") as file:  
        file.seek(0,0)  
        data = file.read(36)  
        data_unpack = struct.unpack("i 24s 8s",data)  
        print("Size: " +str( data_unpack[0]))  
        print("Date: " + data_unpack[1].decode())  
        print("Random: " + data_unpack[2].decode())  
    file.close()
```

Sebastian Solares  
202004822