

Sección: D06

Profesor: MICHEL EMANUEL LOPEZ FRANCO

Tema: Estatus

Ciclo: 2024A

Introducción

Dentro de la computación y la informática al menos en el uso de lo que es software siempre tenemos presente la existencia de las tareas en el sistema operativo de Windows pero no las demás pestañas cuando se abre el administrador de tareas por lo cual da lugar a que se hable precisamente de los servicios que monitorizan el uso de las aplicaciones.

Contenido

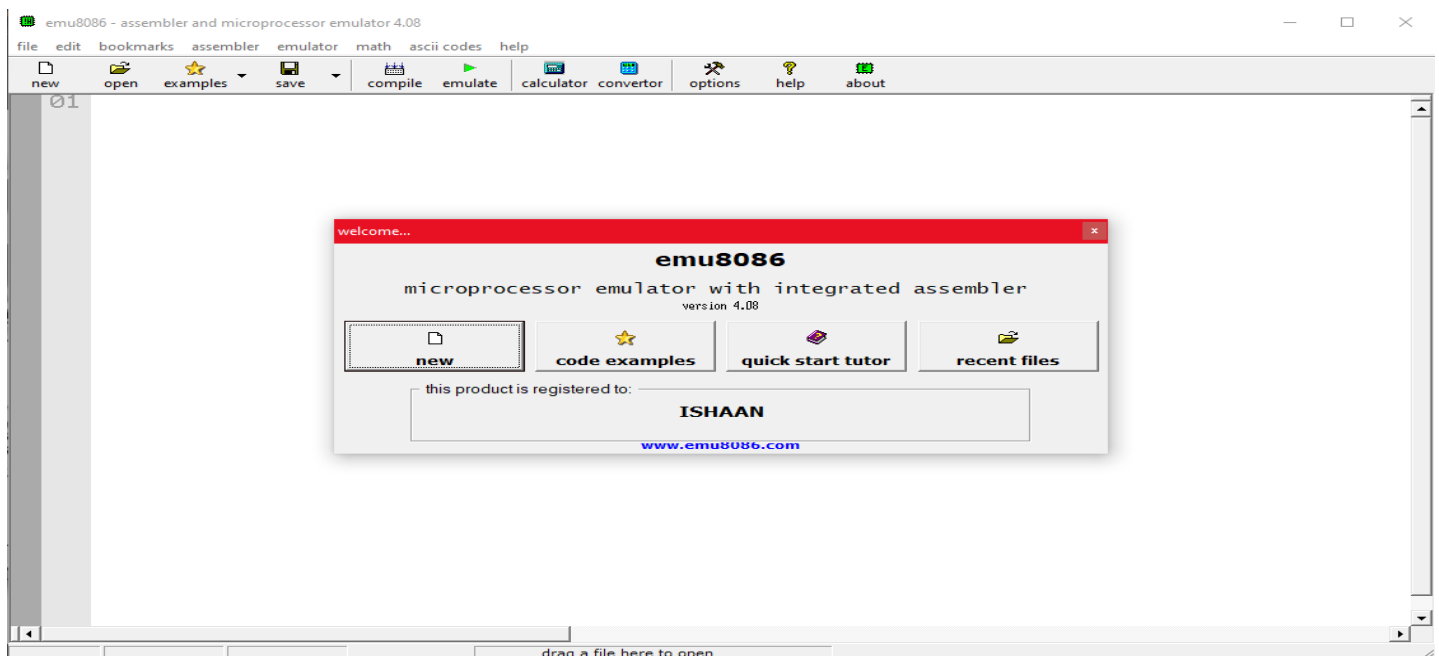
Dentro de las expectativas se quería crear un servicio sin tener que depender del uso de un intérprete de lenguaje para que este funcionara, a como se mostrara a continuación en el siguiente código.

```

1  import os # Importamos módulo os para funciones de sistema operativo
2  import time # Importamos módulo time para hacer delays
3  import psutil # Importamos psutil para obtener información de procesos
4
5  NOMBRE_APP = "emu8086.exe" # Creamos constante con el nombre de la aplicación objetivo en este caso la aplicación de emu8086
6
7  while True:
8      def check_app_status(): # Definimos la función de monitoreo
9
10         for proc in psutil.process_iter(): # Iteramos sobre procesos ejecutándose
11
12             if proc.name() == NOMBRE_APP: # Comparamos nombre de proceso
13
14                 app_pid = proc.pid # Obtenemos ID de Proceso
15                 app_proc = psutil.Process(app_pid) # Objeto psutil.Process
16
17                 cpu_uso = app_proc.cpu_percent(interval=1) # Porcentaje de uso de CPU
18                 mem_uso = app_proc.memory_percent() # Porcentaje de uso de RAM
19
20                 # Mostramos información del estado
21                 print(f"{NOMBRE_APP} corriendo. PID: {app_pid}. CPU: {cpu_uso}%. MEM: {mem_uso}%")
22
23                 return True # Retornamos True porque la encontramos
24
25         print(f"{NOMBRE_APP} no se está ejecutando") # No se encontró el proceso
26         return False # Se retorna False

```

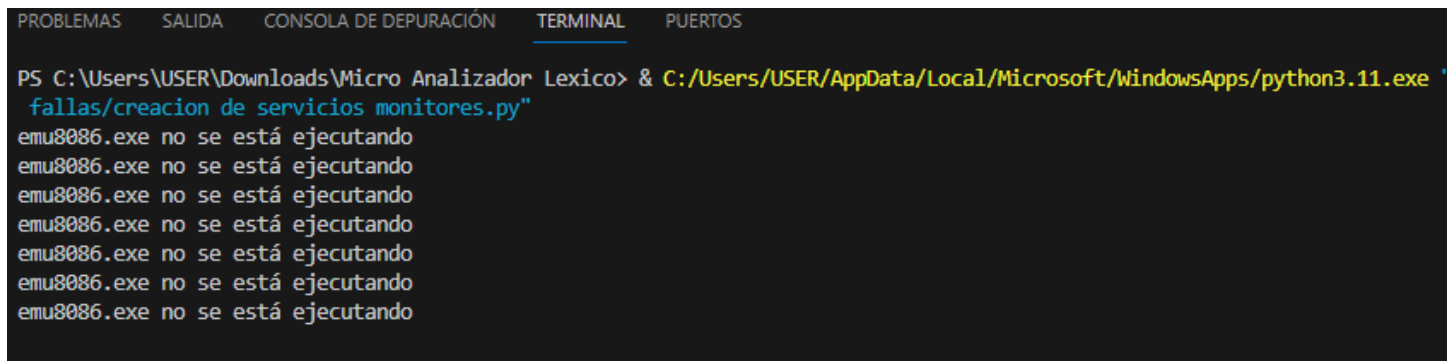
Como se logra observar en este pedazo de código podemos observar que es un código sencillo ya que existen librerías para obtener información de las aplicaciones que están en ejecución o al menos las que se especifican como en este caso es "emu8086.exe" para consiguiente es la siguiente aplicación.



La siguiente parte del código es lo que nos mostrara en consola si está activa o no esta aplicación dado que dará un intervalo de 5 segundos para el monitoreo.

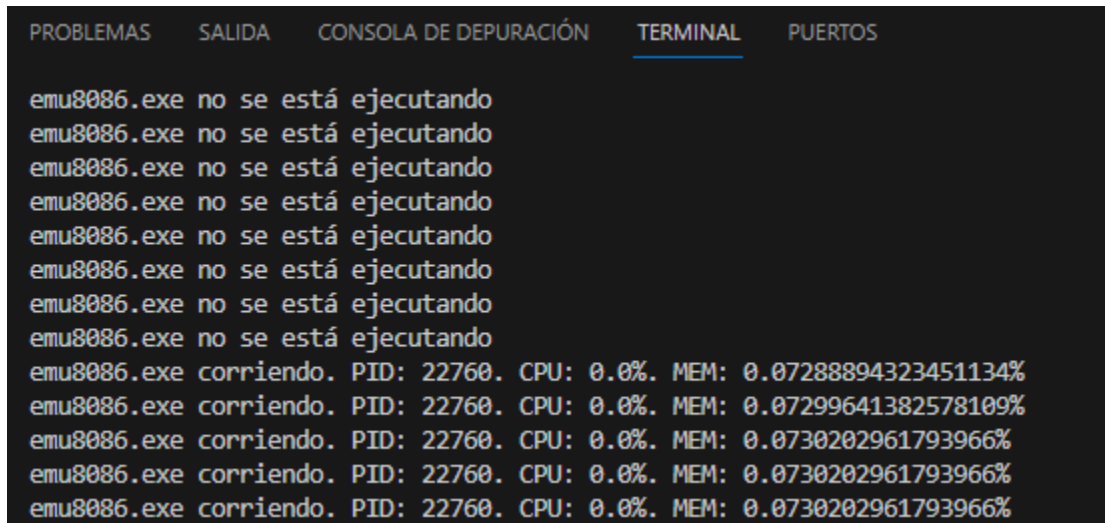
```
25     print(f"{NOMBRE_APP} no se está ejecutando") # No se encontró el proceso
26     return False # Se retorna False
27
28
29     if __name__ == "__main__": # Punto de entrada como programa principal
30
31         while True:
32             check_app_status() # Ejecutamos la monitorización
33             time.sleep(5) # Pausa de 5 segundos
```

Como se muestra en la siguiente captura una vez iniciado el programa se muestra lo siguiente:



```
PS C:\Users\USER\Downloads\Micro Analizador Lexico> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
fallas/creacion de servicios monitores.py"
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
```

Una vez abierta la aplicación, se muestra la siguiente captura donde este programa captura los datos de la aplicación abierta:



```
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe no se está ejecutando
emu8086.exe corriendo. PID: 22760. CPU: 0.0%. MEM: 0.07288894323451134%
emu8086.exe corriendo. PID: 22760. CPU: 0.0%. MEM: 0.07299641382578109%
emu8086.exe corriendo. PID: 22760. CPU: 0.0%. MEM: 0.0730202961793966%
emu8086.exe corriendo. PID: 22760. CPU: 0.0%. MEM: 0.0730202961793966%
emu8086.exe corriendo. PID: 22760. CPU: 0.0%. MEM: 0.0730202961793966%
```

Referencias

- tecnobillo. (s. f.). *Crear Servicios para Windows con Python*. Tecnobillo.

<https://tecnobillo.com/sections/python-en-windows/servicios-windows-python/servicios-windows-python.html>