

Materia: Computación Tolerante a Fallas.

NRC: 179961

Maestro: Lopez Franco, Michael Emanuel

Aula: X-02

Sección: D06

Alumno: Zashuvath López Moreno, Ethan Israel

Código: 216493953



Índice

Introducción	3
Objetivo:	3
Contenido.....	3
Estructura.....	3
Resultados:.....	5

Introducción

Generar un programa utilizando hilos, procesos, demonios y concurrencia.

Objetivo:

Elaborar un programa que tenga concurrencia, hilos, demonios y procesos.

Contenido

A continuación mostrare el programa desglosado y también los resultados obtenidos por el mismo.

Estructura

- 1) Importación de módulos:

```
1  import os
2  import threading
3  import multiprocessing
4  import time
5
```

- 2) Funcion "write_to_file":

```
6  # Función para escribir en un archivo de texto
7  def write_to_file(filename, text):
8      with open(filename, 'w') as file:
9          file.write(text)
10
```

- 3) Funcion "create_file_thread":

```
11 # Función para crear un archivo de texto en un hilo
12 def create_file_thread(file_num):
13     filename = f'file{file_num}.txt'
14     text = f'Este es el archivo {file_num}\n'
15     write_to_file(filename, text)
16
```

- 4) Funcion "create_file_process":

```
17 # Función para crear un archivo de texto en un proceso
18 def create_file_process(file_num):
19     filename = f'file{file_num}.txt'
20     text = f'Este es el archivo {file_num}\n'
21     write_to_file(filename, text)
22
```

- 5) Bloque "if __name__ == '__main__'":

```
23 if __name__ == "__main__":
```

6) Hilos para crear archivos

```
24 # Usar hilos para crear archivos de texto
25 threads = []
26 for i in range(1, 5):
27     thread = threading.Thread(target=create_file_thread, args=(i,))
28     threads.append(thread)
29     thread.start()
30
31 for thread in threads:
32     thread.join()
33
34 print("Archivos de texto creados en hilos")
35
```

7) Procesos para crear archivos:

```
36 # Usar procesos para crear archivos de texto
37 processes = []
38 for i in range(1, 5):
39     process = multiprocessing.Process(target=create_file_process, args=(i,))
40     processes.append(process)
41     process.start()
42
43 for process in processes:
44     process.join()
45
46 print("Archivos de texto creados en procesos")
47
```


8) Creación de un demonio:


```
48 # Crea un demonio para esperar un tiempo antes de que el programa termine
49 def daemon_function():
50     time.sleep(5)
51     print("El demonio ha terminado")
52
53 daemon_thread = threading.Thread(target=daemon_function, daemon=True)
54 daemon_thread.start()
55
```






9) Espera del programa principal:

```
56 # El programa principal continúa mientras el demonio se ejecuta en segundo plano
57 while daemon_thread.is_alive():
58     pass
59
60 print("Programa principal ha terminado")
61
```

Resultados:

Nombre	Fecha de modificación	Tipo	Tamaño
 Programa_Prueba.py	11/09/2023 06:33 p. m.	Python File	2 KB

			
--	--	--	--

Nombre	Fecha de modificación	Tipo	Tamaño
 file1.txt	11/09/2023 08:07 p. m.	Documento de te...	1 KB
 file2.txt	11/09/2023 08:07 p. m.	Documento de te...	1 KB
 file3.txt	11/09/2023 08:07 p. m.	Documento de te...	1 KB
 file4.txt	11/09/2023 08:07 p. m.	Documento de te...	1 KB
 Programa_Prueba.py	11/09/2023 06:33 p. m.	Python File	2 KB