



Computación tolerante a fallas

Universidad de Guadalajara

Nombre: Marco Aurelio Domínguez Amezcua

Código: 216818534

Carrera: Ingeniería en computación.

Semestre: 6to Semestre

Objetivo:

El objetivo de esta práctica es hacer un programa el cual utilice hilos, multiprocesos, demonios o concurrencia, y, además, aprender a manipularlos.

Desarrollo:

Para el desarrollo de esta práctica decidí investigar acerca de que son los hilos, me di cuenta que son muy útiles para la ejecución de programas que contengan múltiples procesos, ya que mientras un hilo realiza un proceso, si es que se tiene más de estos, se puede optimizar el trabajo, ya que el trabajo se realiza más rápido mientras más hilos haya.

Quise crear un programa el cual nos calculara los días faltantes para algunas fechas importantes en el año, decidí elegir los días donde se celebra la navidad y Halloween, que son épocas muy diferentes, pero de igual manera se festejan. El programa lo realice con Python utilizando las librerías de “time”, “threading” y “datetime”.

```
5  ✓ from datetime import *
6    from time import sleep, perf_counter
7    import threading
8
```

1. Librerías importadas

Primeramente, hice una función la cual actualizará el año correspondiente iniciando desde el año actual, utilizando la herramienta Try-Except descartamos que si el año es bisiesto nos ingresen una fecha equivocada, así que el programa no podrá detenerse por ese posible error.

```
9  def add_years(start_date, years):
10     try:
11         return start_date.replace(year=start_date.year + years)
12     except ValueError:
13         # 📌 preserve calendar day (if Feb 29th doesn't exist, set to 28th)
14         return start_date.replace(year=start_date.year + years, day=28)
15
```

2. Función de actualizar fecha

La siguiente función es la que se encarga de calcular los días que faltan desde la fecha actual hasta la fecha festiva de cada año, identificando los años bisiestos y con un tiempo de espera de 1 segundo (esto es opcional, para las ejecuciones posiblemente sean removidas), para después mostrar el mensaje esperado.

```
16 def hoyanavidad(hoy, navidad, bis):
17     bisYears = 0
18     for i in range(0, 6):
19         currentYear = add_years(navidad, i) # año actual = 2023
20
21         for b in range(len(bis)):
22             if bis[b] != currentYear.year:
23                 if b == len(bis)-1:
24                     dias = (currentYear-hoy).days
25                 else:
26                     pass
27             else:
28                 if b == len(bis)-1:
29                     dias = (currentYear-hoy).days
30                     bisYears+=1
31                     dias=dias+bis
32                 else:
33                     pass
34         sleep(1)
35         print(f" 🎄 Faltan {dias} días para NAVIDAD del año {currentYear.year}")
36
```

3. Calcular días para navidad

La siguiente función es similar, solamente que los parámetros son distintos y la festividad también cambia.

```
42 def hoyahalloween(hoy, hw, bis):
43     #start_time = perf_counter()
44     bisYears = 0
45     for i in range(0, 6):
46         currentYear = add_years(hw, i) # año actual = 2023
47
48         for b in range(len(bis)):
49             if bis[b] != currentYear.year:
50                 if b == len(bis)-1:
51                     dias = (currentYear-hoy).days
52                 else:
53                     pass
54             else:
55                 if b == len(bis)-1:
56                     dias = (currentYear-hoy).days
57                     bisYears+=1
58                     dias=dias+bis
59                 else:
60                     pass
61         sleep(1)
62         print(f" 🎃 Faltan {dias} días para HALLOWEEN del año {currentYear.year}")
```

4. Calcular días para Halloween

Para poder poner en uso todas nuestras funciones debemos declarar nuestra fecha de hoy y las fechas de las festividades. Posteriormente, crear un hilo para cada función ya que se van a ejecutar simultáneamente, esto quiere decir que no llevan un orden, sino que se va a mostrar en pantalla quien va haciendo su proceso primero, además, cada hilo tendrá que recibir su objetito (target) que indica que función va a realizar y los argumentos (args) que requiere la función, ya teniendo todo esto establecido iniciamos la ejecución de los hilos (start) y con la función “join()” damos a entender que lo que hay debajo de estos procesos se ejecuta al finalizarlos, en caso que no se ponga podemos continuar leyendo el código.

```
64 start_time = perf_counter()
65
66 bisiestos = [2024, 2028, 2032]
67
68 fechahoy = date.today()
69 navidad = date(2023, 12, 25)
70 halloween = date(2023, 10, 31)
71
72
73 hilo_navidad = threading.Thread(target=hoyanavidad, args=(fechahoy, navidad, bisiestos,))
74 hilo_halloween = threading.Thread(target=hoyahalloween, args=(fechahoy, halloween, bisiestos,))
75 hilo_halloween.start()
76 hilo_navidad.start()
77
78 end_time = perf_counter()
79
80 print(f'\nIt took {end_time- start_time: 0.2f} second(s) to complete.')
```

Resultado

Como nos podemos dar cuenta, no depende de cual hilo se ejecuta primero, para terminar antes, en este ejemplo el caso “Navidad” empezó despues que el caso “Halloween”, pero terminó antes, aquí nos podemos percatar de la utilidad de esta herramienta.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK: VARIABLES

te a fallas - Michel\PythonCodes> & C:/Python311/python.exe "c:/Uste a fallas - Michel\Py
te a fallas - Michel\PythonCodes> & C:/Python311/python.exe "c:/Users/Marco Aurelio/OneDr
ive/Escritorio/6to semestre/Tolerante a fallas - Michel/PythonCodes/navidadfuncional.py"
🎃 Faltan 51 días para HALLOWEEN del año 2023
🌲 Faltan 106 días para NAVIDAD del año 2023
🎃 Faltan 417 días para HALLOWEEN del año 2024
🌲 Faltan 472 días para NAVIDAD del año 2024
🌲 Faltan 837 días para NAVIDAD del año 2025
🎃 Faltan 782 días para HALLOWEEN del año 2025
🎃 Faltan 1147 días para HALLOWEEN del año 2026
🌲 Faltan 1202 días para NAVIDAD del año 2026
🌲 Faltan 1567 días para NAVIDAD del año 2027
🎃 Faltan 1512 días para HALLOWEEN del año 2027
🎃 Faltan 1878 días para HALLOWEEN del año 2028
🌲 Faltan 1933 días para NAVIDAD del año 2028

It took 6.01 second(s) to complete.
```

En caso de borrar la función “join()” podemos ver que el código se sigue ejecutando sin importar que los hilos no hayan terminado. Como en el siguiente ejemplo:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK: VARIABLES

Michel/PythonCodes"
PS C:\Users\Marco Aurelio\OneDrive\Escritorio\6to semestre\Tolerante a fallas - Michel\Pyt
honCodes> & C:/Python311/python.exe "c:/Users/Marco Aurelio/OneDrive/Escritorio/6to semest
re/Tolerante a fallas - Michel/PythonCodes/navidadfuncional.py"
Este código está debajo de los hilos
🎃 Faltan 51 días para HALLOWEEN del año 2023
🌲 Faltan 106 días para NAVIDAD del año 2023
🌲 Faltan 472 días para NAVIDAD del año 2024
🎃 Faltan 417 días para HALLOWEEN del año 2024
🌲 Faltan 837 días para NAVIDAD del año 2025
🎃 Faltan 782 días para HALLOWEEN del año 2025
🎃 Faltan 1147 días para HALLOWEEN del año 2026
🌲 Faltan 1202 días para NAVIDAD del año 2026
🌲 Faltan 1567 días para NAVIDAD del año 2027
🎃 Faltan 1512 días para HALLOWEEN del año 2027
🎃 Faltan 1878 días para HALLOWEEN del año 2028
🌲 Faltan 1933 días para NAVIDAD del año 2028
```

Conclusión

El usar diferentes de herramientas es muy interesante porque te das cuenta que puedes resolver problemas de formas más sencillas o más eficaces, en el caso de los hilos y multiprocesos me di cuenta que a veces no tenemos considerado el utilizar el tiempo que tarda una tarea en ejecutarse de manera correcta, sino que nos esperamos a que esa termine para continuar con la siguiente, pero en algunos casos vale más aprovechar ese pequeño lapso de tiempo para ejecutar más procesos. Siento que este tema fue muy útil y quiero aprender más de él.

Bibliografía

- Python threading. (2020, December 17). Python Tutorial - Master Python Programming For Beginners from Scratch; Python Tutorial.
<https://www.pythontutorial.net/python-concurrency/python-threading/>
- Follow, Y. (2019, April 18). time.perf_counter() function in Python. GeeksforGeeks. https://www.geeksforgeeks.org/time-perf_counter-function-in-python/
- NeuralNine [@NeuralNine]. (2023, January 10). Python Threading Explained in 8 Minutes. Youtube.
https://www.youtube.com/watch?v=A_Z1lgZLSNc