UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

Facultad de Ingeniería

Ingeniería en Ciencias de la Computación

COMPUTO PARALELO Y DISTRIBUÍDO

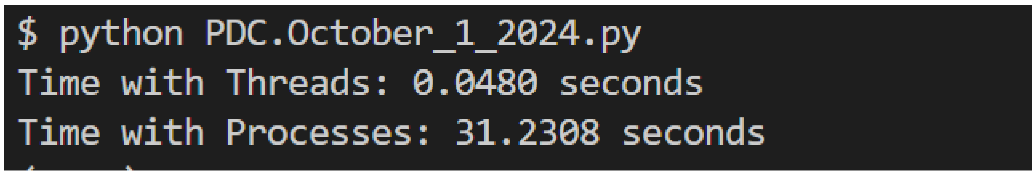
1.42 Actividad 13: Procesos en Python

Alumna: ADRIAN (ADORA) GONZÁLEZ DOMÍNGUEZ *[359834]*

Asesor: JOSE SAUL DE LIRA MIRAMONTES

1 de octubre de 2024

```python
import threading
import multiprocessing
import random
import time
import numpy as np

# Function to fill a row with random values
def fill_row(matrix, row):
    matrix[row] = [random.random() for _ in range(len(matrix[row]))]

def fill_matrix_with_threads(matrix):
    threads = []
    for row in range(len(matrix)):
        thread = threading.Thread(target=fill_row, args=(matrix, row))
        threads.append(thread)
        thread.start()

    for thread in threads:
        thread.join()

def fill_matrix_with_processes(matrix):
    processes = []
    for row in range(len(matrix)):
        process = multiprocessing.Process(target=fill_row,
args=(matrix, row))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()

def compare_processing_times(matrix_size):
    matrix_threads = np.zeros((matrix_size, matrix_size))
    matrix_processes = np.zeros((matrix_size, matrix_size))

    start_threads = time.time()
    fill_matrix_with_threads(matrix_threads)
    end_threads = time.time()
    time_threads = end_threads - start_threads
    start_processes = time.time()
    fill_matrix_with_processes(matrix_processes)
    end_processes = time.time()
    time_processes = end_processes - start_processes

    print(f"Time with Threads: {time_threads:.4f} seconds")
    print(f"Time with Processes: {time_processes:.4f} seconds")

if __name__ == "__main__":
    matrix_size = 200  # Define the size of the matrix (200x200)
    compare_processing_times(matrix_size)
```



```
$ python PDC.October_1_2024.py
Time with Threads: 0.0480 seconds
Time with Processes: 31.2308 seconds
```