

COMPUTACION PARALELA

Practica Computacion Paralela - I

Jhosett Russbell Challco Baez

0.1 EJERCICIO 1

El proceso de cifrado de un sistema de contraseñas está subdividido en dos (02) partes x1(SHA-256),x2(MD5) iterativamente y secuencialmente, se asume que cada una de las partes es independiente, además también se conoce que x2 consume el 35% de la computación. Si tenemos 02 opciones de mejora:

F1 = Optimización de x1 : 8 veces más rápido

F2 = Optimización de x2 : 3 veces más rápido

¿Cuál es la más óptima y por qué?

RESULTADOS:

X	100%
x1	65%
x2	35%

Vel.x1	8
Vel.x2	3

F1	43.13%
F2	76.67%

0.2 EJERCICIO 2

Se requiere una mejora en la velocidad de un CPU mediante un procedimiento de overclocking pero esta mejora solo impactaría en procesadores con una velocidad mayor a 3.2GHz y en un 20 porciento ¿Calcule la aceleración para los siguientes casos?

- a) Procesador A : 2.86 GHz
- b) Procesador A : 3.58 GHz
- c) Procesador A : 3.14 GHz
- d) Procesador A : 3.81 GHz

CODIGO: (L-L)

```
import time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def velocidad():
    time_ini = time.time()
    NV = [2.86, 3.58, 3.14, 3.81]
    for i in NV:
        if i > 3.2:
            M = i * 1.2
            print(M)
        else:
            print(i)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#sin hilos
velocidad()
print("")

#NOTA DE RESULTADO:
# 2.86
# 3.58*1.2=4.296
# 3.14
# 3.81*1.2=4.572
```

RESULTADOS: (L-L)

```
'C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\JHOSSETT\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50019' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg02 (24_10_22).py'
2.86
4.296
3.14
4.572
TIEMPO: 0.002656698226928711
```

CODIGO: (L-H)

```
import time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def velocidad():
    time_ini = time.time()
    NV = [2.86, 3.58, 3.14, 3.81]
    for i in NV:
        if i > 3.2:
            M = i * 1.2
            print(M)
        else:
            print(i)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#con hilos
t = Thread(target=velocidad, args=())
t.start()
print("")

#NOTA DE RESULTADO:
# 2.86
# 3.58*1.2=4.296
# 3.14
# 3.81*1.2=4.572
```

RESULTADOS: (L-H)

```
'C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\JHOSSETT\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50057' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg02 (24_10_22).py'

2.86
4.296
3.14
4.572
TIEMPO: 0.002138376235961914
```

CODIGO: (L-P)

```
import time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def velocidad():
    time_ini = time.time()
    NV = [2.86, 3.58, 3.14, 3.81]
    for i in NV:
        if i > 3.2:
            M = i * 1.2
            print(M)
        else:
            print(i)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#procesos
if __name__ == '__main__':
    multiprocessing.set_start_method('spawn')
    p = Process(target=velocidad, args=())
    p.start()

#NOTA DE RESULTADO:
# 2.86
# 3.58*1.2=4.296
# 3.14
# 3.81*1.2=4.572
```

RESULTADOS: (L-P)

```
lib\python\debugpy\adapter/../..\debugpy\launcher' '49368' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg02 (24_10_22).py'
2.86
4.296
3.14
4.572
TIEMPO: 0.007721424102783203
PS C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel>
```

0.3 EJERCICIO 3

Realizar una funcion que calcule la distancia entre dos puntos en una implementacion lineal y en una implementacion utilizando hilos, asi mismo evaluar la diferencia entre estas. $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

CODIGO: (L-L)

```
import time #pip install time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def distancia():
    time_ini = time.time()
    x1=1
    y1=1
    x2=3
    y2=3
    d=math.sqrt(((x2-x1)**2)+((y2-y1)**2))
    x=(x2-x1)
    y=(y2-y1)
    print ("Distancia entre 2 puntos: ", d)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#sin hilos
distancia()
```

RESULTADOS: (L-L)

```
'C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\JHOSSETT\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49735' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg03 (24_10_22).py'
Distancia entre 2 puntos:  2.8284271247461903
TIEMPO:  0.000099945068359375
PS C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel>
```

CODIGO: (L-H)

```
import time #pip install time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def distancia():
    time_ini = time.time()
    x1=1
    y1=1
    x2=3
    y2=3
    d=math.sqrt(((x2-x1)**2)+((y2-y1)**2))
    x=(x2-x1)
    y=(y2-y1)
    print ("Distancia entre 2 puntos: ", d)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#hilos
t = Thread(target=distancia, args=())
t.start()
```

RESULTADOS: (L-H)

```
'C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\JHOSSETT\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49607' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg03 (24_10_22).py'
Distancia entre 2 puntos:  2.8284271247461903
TIEMPO:  0.0010695457458496094
PS C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel>
```

CODIGO: (L-P)

```
import time #pip install time
import math
from threading import Thread
import multiprocessing
from multiprocessing import Process

def distancia():
    time_ini = time.time()
    x1=1
    y1=1
    x2=3
    y2=3
    d=math.sqrt(((x2-x1)**2)+((y2-y1)**2))
    x=(x2-x1)
    y=(y2-y1)
    print ("Distancia entre 2 puntos: ", d)
    time_end = time.time()
    total = time_end - time_ini
    print("TIEMPO: ", total)

#procesos
if __name__ == '__main__':
    multiprocessing.set_start_method('spawn')
    p = Process(target=distancia, args=())
    p.start()
```

RESULTADOS: (L-P)

```
'C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\JHOSSETT\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49821' '--' 'c:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel\Practica_Preg03 (24_10_22).py'
Distancia entre 2 puntos:  2.8284271247461903
TIEMPO:  0.010359525680541992
PS C:\Users\JHOSSETT\AppData\Local\Programs\Python\Python310\Lib\site-packages\parallel>
```