UNIVERSIDAD NACIONAL DEL ALTIPLANO

listings *Facultad de Ingenieria Estadistica e Informatica*

COMPUTACIÓN PARALELA

PRÁCTICA - I

Presentado por:
Grylia Yaneth Chata Iscarra
Código: 190496
Semestre: VIII - Unit II
Email: gchatai@est.unap.edu.pe

Docente: Ing:
TORRES CRUZ FRED

27 de octubre de 2022

listings

Índice

xcolor

1. Ejercicio.

1.1. Solución: Mediante la ley de Amdahl

$$x = x_1 + x_2$$
$$x = 65\,\% + 35\,\%$$
$$f_1 = 65\,\% \div 8 + 35\,\% = 43{,}1\,\%$$
$$f_2 = 65\,\% + 35\,\% \div 3 = 76{,}7\,\%$$

$$RPTA : Lo\ más\ F1\ es\ 8\ veces\ 43{,}1\,\%\ más\ rápido$$

2. Ejercicio.

2.1. Solución: Si son mayores que 3.2 se saca el 20 % y sino se mantiene

$$Procesador A : 2{,}86GHZ = 2{,}86GHZ$$

$$Procesador A : 3{,}58GHZ = 3{,}58 * 1{,}2 = 4{,}296GHZ$$

(Sacando el 20 %)

$$Procesador A : 3{,}14GHZ = 3{,}14GHZ$$

$$Procesador A : 3{,}81GHZ = 3{,}81 * 1{,}2 = 4{,}5726GHZ$$

(Sacando el 20 %)

2.2. Solución: Serie función

2.3. Codigo.

```
Import time
#pip install time
#from threading import Thread


def hola mundo():
    time ine = time.time()
    print("Inicio_")
    time.sleep(1)


    ghz=float(input("ingrese procesador: "))
    if (ghz<3.2):
        print("No se puede optimizar: ")
    else:
        print("Se optimizo en: ", ghz*1.2)

    print ("Fin_")
    time_end = time.time()
    total = time_end - time_ini
    print(total)

for i in range(1):
    holamundo()
```

2.4. Impresión.

IMGRESANDO: 2.86
Inicio
Ingrese procesador: 2.86
No se puede optimizar:
Fin
58.6339366436004 64
INGRESANDO: 3.58
Inicio
Ingrese procesador: 3.58
Se optimizo en: 4.296

Fin
6.717825889587402
INGRESANDO: 3.14
Inicio
Ingrese procesador: 3.14
No se puede optimizar:
Fin
17.0236713886261
INGRESANDO: 3.81
Inicio
Ingrese procesador: 3.81
Se optimizo en: 4.572
Fin
15.939423084259033

2.5. Solución: Serie threads

2.6. Codigo.

```
import time
#pip install time
from threading import Thread

def Ejer2():
    time_ini = time.time()
    print("Inicio_")
    time.sleep(1)

    ghz=float(input("Ingrese procesador: "))
    if (ghz<3.2):
        print("No se puede obtimizar: ")
    else:
        print("Optimizacion: ", ghz*1.2)

    print ("Fin_")
    time_end = time.time()
    total = time_end - time_ini
    print(total)
```

```
20
21 for i in range(1):
22     t = Thread(target=Ejer2, args=())
23     t.start()
```

2.7. Impresión.

IMGRESANDO: 2.86
Inicio
Ingrese procesador: 2.86
No se puede optimizar:
Fin
102.10373020172119
INGRESANDO: 3.81
Inicio
Ingrese procesador: 3.81
Se optimizo en: 4.572
Fin
15.250174045562744

2.8. Solución: Procesos

2.9. Codigo.

```
1  import time
2  import multiprocessing
3
4  from multiprocessing import Process
5
6  def increase():
7      time_ini = time.time()
8      print("_Inicio_")
9      #Proceso
10     i = 0
11     for _ in range(1000000000):
12         i += 1
13     print(i)
14     #/Proceso
```

```
15      print ("_Fin_")
16      time_end = time.time()
17      total = time_end - time_ini
18      print(total)
19
20  if __name__ == '__main__':
21      multiprocessing.set_start_method('spawn')
22      for _ in range(5):
23          p = Process(target=increase, args=()) #Proceso
24          p.start()
25      # increase()
```

2.10. Impresión.

IMGRESANDO: 2.86
Inicio
Ingrese procesador: 2.86
No se puede optimizar:
Fin
58.463936643600464

3. Ejercicio.

3.1. Solución: Distancia

3.2. Codigo.

```
1  import math
2  import time #pip install time
3  from threading import Thread
4  import math
5  def n2(n):
6      time_ini = time.time()
7      print("Inicio_")
8
9      print("Ingrese los valores de la coordenada: ")
10
11     x1=int(input("Ingrese el valor x1: "))
```

```
12      y1=int(input("Ingrese el valor y1: "))
13
14      print
15      x2=int(input("Ingrese el valor x2: "))
16      y2=int(input("Ingrese el valor y2: "))
17
18      distancia =math.sqrt((x2-x1)*2+(y2-y1)*2)
19      print (distancia)
20      print ("Fin_")
21      time_end = time.time()
22      total = time_end - time_ini
23      print(total, "\t")
24
25
26  for i in range(1):
27      t1 = Thread(target= n2, args=(3,))
28      t1.start()
```

3.3. Impresión.(Tomando en cuenta los puntos (1.1) y (3.2)

Inicio
Ingrese los valores de la coordenada:
Ingrese el valor x1: 1
Ingrese el valor y1: 1
Ingrese el valor x2: 3
Ingrese el valor y2: 2
2.449489742783178
Fin
18.68965435028076
