

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ARQUITECTURA DE COMPUTADORAS

Laboratorio 12
(2023-2)

Indicaciones generales:

- Materiales permitidos: Wiki del curso, apuntes de clase o documentación de python online.
- **Está prohibido el uso de cualquier modelo de lenguaje** como ChatGPT o Github Copilot. A cualquier alumno que se le detecte que ha consultado un modelo de lenguaje se le pondrá nota 0 (cero) en el laboratorio.
- Usted debe subir a Paideia 1 solo archivo comprimido .zip con el nombre **L12_CODIGO.zip**. Este archivo comprimido debe tener carpetas con nombres 'p1.py' y 'p2.py' con archivos de python para cada pregunta, **además de sus respuestas teóricas en un pdf denominado L12_CODIGO.pdf**
- El horario máximo permitido para subir el archivo es a las 10:00:00 pm. Pasada esa hora, habrá una penalidad de 2 puntos por cada minuto extra que se demore en entregar su archivo.

Puntaje total: 20 puntos

Pregunta 1 (10 pt)

Se desea copiar cierto número de archivos de una carpeta a otra y comparar la performance de una implementación síncrona y asíncrona.

- a) Escriba la función `generar_archivos(M,N)` que genere M archivos de N caracteres cada uno, en una subcarpeta denominada "original". Para ello, solo debe cambiar el nombre de su archivo a `./original/archivo1`. Pruebe su función con $M = 3$ y $N = 1024$. (1 pt)
- b) Escriba una función que copie los archivos de la carpeta "original" a una carpeta llamada "copia" de manera **síncrona**. Solo utilice las funciones nativas de python. (1 pt)
- c) Escriba una función que copie los archivos de la carpeta "original" a una carpeta llamada "copia" de manera **asíncrona**. (2 pt)
- d) Fijando $M = 3$, medir el tiempo de ejecución variando N para potencias de 2 desde 2^{10} hasta 2^{25} . **Puede disminuir estos valores en hasta 5 órdenes de magnitud si en su equipo toma demasiado tiempo.** Guardar los tiempos de ejecución en un arreglo y generar una gráfica de N vs tiempos de ejecución. ¿Cómo afecta el tamaño de archivo al tiempo de ejecución y por qué? (3 pt)
- e) Fijando $N = 2^{20}$, medir el tiempo de ejecución variando M de 2 hasta 10. Guardar los tiempos de ejecución en un arreglo y generar una gráfica de M vs tiempos de ejecución. ¿Cómo afecta el número de archivos al tiempo de ejecución y por qué? (3 pt)

Para generar las gráficas puede utilizar el siguiente código:

```
import matplotlib.pyplot as plt
import time

lista_N = [2**i for i in range(10,26)]
tiempos_sync = []
tiempos_async = []

# Inserte aqui el codigo que haga falta

for N in lista_N:
    # Inserte aqui el codigo que haga falta
    tiempos_sync.append(tiempo_medido)

    # Inserte aqui el codigo que haga falta
    tiempos_async.append(tiempo_medido)

plt.plot(lista_N,tiempos_sync)
plt.plot(lista_N,tiempos_async)
plt.xlabel('File size [bytes]')
plt.ylabel('Copy time [ms]')
plt.legend(['Sync','Async'])
plt.savefig('SizeVsTime.png')
plt.cla()
```

Pregunta 2 (5 pt)

Se desea descargar un conjunto de 29 imágenes, que se pueden encontrar en los siguientes enlaces:

https://raw.githubusercontent.com/SebastianMerino/Threading/main/images/01.png

https://raw.githubusercontent.com/SebastianMerino/Threading/main/images/02.png

...

https://raw.githubusercontent.com/SebastianMerino/Threading/main/images/29.png

- a) Descargue los archivos secuencialmente y mida el tiempo de ejecución. Puede utilizar la función `urlopen()`, la cual funciona de manera similar a la función `open()` de python (1 pt).

```
from urllib.request import urlopen
url = ''

with urlopen(url) as page:
    image_data = page.read() # data como objeto binario
```

- b) Descargue los archivos utilizando hilos. Genere un hilo por cada imagen a descargar. Mida el tiempo de ejecución y comente sus resultados (2 pt).
- c) Descargue los archivos utilizando solo 3 hilos. Mida el tiempo de ejecución. ¿Obtuvo mejores o peores resultados que en b)? ¿Por qué? (2 pt).

Importante: Para la medición de los tiempos de ejecución **en la pregunta 2**, se debe realizar la prueba 5 veces y obtener el tiempo mediano.