

Homework 1: Multitasking

多任務 (Multitasking) 指的是在單核心或多核心的電腦系統上，透過**並發執行** (Concurrent Execution) 或**平行執行** (Parallel Execution) 來達到同時執行多個任務的技術。在本次作業中，要請同學比較 Python 語言底下的**多執行緒** (Multithreading)、**多行程** (Multiprocessing)、**協程** (Coroutine) 幾項技術，在不同任務下執行的效率。

作業項目

➤ 項目 A : Proof of Work

Proof of Work (PoW) 是一種讓 client 端執行複雜運算，而答案能被 server 端快速驗證的機制。在此任務中，請同學執行類似於 PoW 的運算，流程如下：

1. 輸入為一個長度為 5，由不含空白的「ASCII 可顯示字元」(即字碼 0x21~0x7E) 組成的字串 S
16進位(hex)
2. 同學須找出一個同樣長度為 5 的字串 C ，使得 $\text{SHA256}(C||S)$ 的十六位進制表示法中，前 5 位為 0

範例

輸入為 `abcde`，輸出為 `u3x"!abcde`

因為 $\text{SHA256}(\text{u3x"!abcde}) = 0x0000036d66 \dots$

請注意，答案可能不唯一

➤ 項目 B : 獲取網頁標題

在此任務中，輸入為一筆 URL 字串，請同學透過 Python 的 requests 模組來獲取網頁內容，並從中印出該網頁的標題

範例

輸入為 `https://www.nctu.edu.tw`

輸出為 `NCTU 國立交通大學`

作業要求

➤ 程式部分

- 程式的第一個輸入指定要執行的任務，**1** 表示任務一；**2** 表示任務二
- 程式的第二個輸入指定執行任務的方式，**1** 是使用多執行緒；**2** 是使用多行程；**3** 是使用協程。若方式為多執行緒/多行程，會跟著一個整數指定執行緒數/行程數。
- 最後會有一筆整數輸入指定有幾筆任務，後面跟著任務輸入
- 任務一請輸出每一筆字串 **S** 對應的 **C||S**
- 任務二請輸出輸入網址對應的網頁標題
- 最後請輸出任務執行所花的時間
- 輸出的順序無規定
- 請於報告中註明使用的 Python 版本 (建議使用 Python3.6 以上)，若有使用額外套件也請於報告註明
- 檔名格式為：**{student_ID}_hw1.py**

範例

輸入：

1

1 4

1

abcde

輸出：

u3x"!abcde

• 報告部分

對於每個任務，請同學以 100 筆不同輸入來進行以下實驗 (可以使用助教提供的範例輸入)。請紀錄執行時間，且盡量仔細地解釋造成此結果的可能原因 (可多次進行取平均以避免誤差)

檔名格式為：**{student_ID}_hw1_report.pdf**

一、執行緒數量對效能的影響

請分別用 1、2、4、100 個執行緒來執行任務，並且比較執行緒數量分別對兩個任務的影響

二、 行程數對效能的影響

請分別用 1、2、4、100 個行程來執行任務，並且比較行程數量分別對兩個任務的影響

三、 多執行緒、多行程、協程的效能比較

請分別用單執行緒、100 個執行緒、100 個行程、協程來執行任務，並且比較它們的效能關係

參考資料

- [hashlib — Secure hashes and message digests](#)
- [Requests: HTTP for Humans](#)
- [threading — Thread-based parallelism](#)
- [multiprocessing — Process-based parallelism](#)
- [asyncio — Asynchronous I/O](#)