

OS_HW1

1101602 吳宇婕

- **作業需求：**
 - 撰寫一支多執行緒程式，並說明程式以及效果。
- **繳交**
 - pdf文字說明，含程式描述、執行畫面截圖、執行時間分析。
 - 程式碼 (任何程式語言皆可)
- **我的作業前情提要：**
 - 使用語法：Python
 - 亮黃色字體為我的補充說明
 - Code : https://github.com/yuiwuulab/1131_OS_HW.git

作業請至下一頁~

Code

```
12 import threading → 引入 Threading module
13 import time
14
15
16
17 num_time = 0
18 let_time = 0
19
20 def print_sleep(): → Thread 1
21     start_time = time.time() # 記錄開始時間
22     for i in range(10):
23         print(f"目前 : {threading.current_thread()}") → 輸出目前所在的 thread
24         print(f"Number(sleep): {i}")
25         time.sleep(0.5)
26     end_time = time.time() # 記錄結束時間
27     global num_time
28     num_time = end_time - start_time # 執行時間
29
30
31 def print_num(): → Thread 2
32     start_time = time.time() # 記錄開始時間
33     for j in range(10):
34         print(f"目前 : {threading.current_thread()}") → 輸出目前所在的 thread
35         print(f"num: {j}")
36         time.sleep(1)
37     end_time = time.time() # 記錄結束時間
38     global let_time
39     let_time = end_time - start_time # 執行時間
40
```

- 圖片上亮黃色部分為我的補充解釋
- Tread 1 (print_sleep)
 - 功能 : print 出 0 ~ 9 · 設有 sleep (0.5)
- Tread 2 (print_nums)
 - 功能 : print 出 0 ~ 9 · 設有 sleep (1)

Code

```
42 |
43 print(f"目前(main) : {threading.current_thread()}" )
44
45 # 創建兩個執行緒
46 thread1 = threading.Thread(target=print_sleep)
47 thread2 = threading.Thread(target=print_num)
48
49
50
51
52 # 啟動執行緒
53 thread1.start()
54 thread2.start()
55 print(f"| 執行數量 : {threading.active_count()} |" )
56
57
58 thread1.join() # 等待thread1 結束
59 thread2.join() # 等待thread2 結束
60
61 #等兩者結束才會往下執行
62
63 print("-----" )
64 print("All threads finished.")
65 print("兩者時間比較:")
66 print(f"Thread 1 (print_sleep): {num_time} seconds")
67 print(f"Thread 2 (print_num): {let_time} seconds")
68 print(f"Difference: {abs(num_time - let_time)} seconds")
69
```

輸出目前所在的thread

分別創建了兩個執行緒：
Thread1 (print_sleep)
Thread 2 (print_num)

啟動執行緒

查看執行數量

等待兩者執行結束

補充 threading：thread module 中最重要的class之一，用於創建和操作執行緒

輸出

```
目前(main) : <_MainThread(MainThread, started 22704)>
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 0
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
| 執行數量 : 3 |
```

```
num: 0
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 1
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
num: 1
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 2
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 3
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
num: 2
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 4
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 5
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
num: 3
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 6
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 7
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
num: 4
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 8
```

```
目前 : <Thread(Thread-1, started 12752)>
```

```
Number(sleep): 9
```

```
目前 : <Thread(Thread-2, started 15852)>
```

```
num: 5
```

Main 也算是一個thread
(Code 第43行輸出)
所以其實除了設好的
thread 1、2 總共有3個執行緒

每次跑進個別的執行緒中會輸出
正在執行哪一個執行緒

輸出

```
目前 : <Thread(Thread-2, started 15852)>  
num: 6  
目前 : <Thread(Thread-2, started 15852)>  
num: 7  
目前 : <Thread(Thread-2, started 15852)>  
num: 8  
目前 : <Thread(Thread-2, started 15852)>  
num: 9  
-----
```

```
All threads finished.
```

```
兩者時間比較:
```

```
Thread 1 (print_sleep): 5.112163066864014 seconds
```

```
Thread 2 (print_num): 10.08767819404602 seconds
```

```
Difference: 4.975515127182007 seconds
```

```
PS D:\OneDrive - 元智大學\大學專用\1122\OS\hw1>
```

因為程式碼第59行
thread2.join()，所以後續在
main中的print結果這些東西會
在 thread2 後執行

以上是我的作業，謝謝助教、教授～