

# 《資料處理》

一、假設有三個行程 (Process) 在很短的時間內依序先後抵達等待佇列，每個行程所需CPU 執行的時間如下表：(25分)

行程抵達次序	所需執行時間 (毫秒)
P1	6
P2	3
P3	8

試問若作業系統採用以下三種不同的排程演算法，每個行程所需的平均等待時間分別為何？(1)先來先做排程 (First-Come, First-Served)；(2)循環分配排程 (Round Robin) 並以3毫秒為時間切割額度 (Time Slice)；(3)假設P1, P2, P3同時抵達等待佇列，最短工作優先排程 (Shortest Job First)。(25分)

## 試題評析

這題是比較少見的作業系統考題，嚴格上來說，對於資處的考科考作業系統可能有些太專業了，雖然考的並非很複雜的概念，但需要具備相關作業系統理論知識才比較能夠作答。

答：

- (1)FCFS: 平均等待時間 =  $(0+6+3)/3 = 5ms$   
 (2)Round Robin: 平均等待時間 =  $[(12-6)+(6-3)+(17-8)]/3 = 6ms$   
 (3)SJF: 平均等待時間 =  $[(9-6)+(3-3)+(17-8)]/3 = 4ms$

二、關聯代數 (Relational Algebra) 係用來描述關聯資料庫中資料表之間的操作方式，今有兩個關聯資料表 Author 及 Book，試將以下關聯代數運算式轉換成適當之 SQL 語法。(25分)

$$\text{Result} = \pi_{\text{Author.Name, Book.Title}} (\sigma_{\text{Book.Category} = \text{'科幻'}} (\sigma_{\text{Author.id=Book.Aid}} (\text{Author} \times \text{Book})))$$

## 試題評析

非常典型的高考關聯式代數考題，考的內容其實不難，比較像是再考驗同學對於關聯式代數的熟悉程度而已

## 考點命中

《高點·高上資料處理講義》第二回，唐箏編撰，頁69-70。

答：

下列兩解皆可：

- (1)SELECT Author.Name, Book.Title  
 FROM Author, Book  
 WHERE Author.id=Book.Aid and Book.Category='科幻'  
 (2)SELECT Author.Name, Book.Title  
 FROM Author JOIN Book ON Author.id=Book.Aid  
 WHERE Book.Category='科幻'

三、請說明以下三種無線通訊技術在傳輸距離、傳送速率、耗電量、成本等面向之差別及使用情境。(1)ZigBee (2)Wi-Fi (3)Bluetooth。(25分) (25分)

## 試題評析

這題完全是考記憶的題目，其中Wi-Fi與Bluetooth是較為常見的技術，同學應該比較容易回答，zigbee則是比較少見的例子，這部分比較仰賴資訊通識的能力

## 考點命中

《高點·高上資料處理講義》第一回，唐箏編撰，頁80藍芽與頁84 WIFI。

答：

	ZigBee	Wi-Fi	Bluetooth
傳輸距離	50-300m	100-300m	1-10m

傳輸速率	250kbps	300Mbps	24Mbps
功耗	5mA	10-50mA	<30mA
成本	低	高	中
應用範圍	自動化設備、遙控裝置、無線設備，多使用於工業化廠房	任何可連接無線網路的裝置，如平板電腦、手機、筆電等	適用於短距離無線連接裝置，如藍芽耳機、音響、汽車系統、滑鼠、鍵盤等

四、請解釋以下資料處理技術的運作原理。(1)合併排序法 (Merge Sort) (2)貪婪演算法 (Greedy Algorithm) (25分)

試題評析	這題是資料結構與演算法的試題，其中合併排序法因為比較常見所以同學應較容易作答，貪婪演算法其實是屬於演算法的問題，是比較偏電腦科學(資工)取向的考題，因此難度可能稍微高一些。
考點命中	《高點·高上資料處理講義》第四回，唐箏編撰，頁37。

答：

(一)合併排序法：

為一種Divide-and-Conquer的演算法，使用遞迴方式執行，分為分割與合併兩個步驟，做法如下：

1.分割：

- (1)將原始陣列平分為兩個較小的子陣列
- (2)將較小的兩子陣列再分別平分
- (3)遞迴執行直到每個小陣列元素只剩一個

2.合併：

- (1)排序兩個只有單一元素的陣列並合併成小陣列
- (2)將兩個由1.產生的小陣列排序並合併成更大的陣列
- (3)遞迴執行直到合併為原始陣列大小

演算法：假設排序陣列為A，起始位置p至終點位置r

Merge(A,p,r)

if(p<r)

q=(p+r)/2

Merge(A,p,q)

Merge(A,q+1,r)

Merge\_sort(A,p,q,r)

Merge\_sort(A,p,q,r)

$n_1 = q - p + 1$

$n_2 = r - q$

Let L[1... $n_1$ +1] and R[1... $n_2$ +1] be new array

for i=1 to  $n_1$

L[i] = A[p+i-1]

for j = 1 to  $n_2$

R[j] = A[q+j]

L[ $n_1 + 1$ ] =  $\infty$

R[ $n_2 + 1$ ] =  $\infty$

i,j=1

for k = p to r

if(L[i] ≤ R[j])

A[k]=L[i]

i = i+1

【版權所有，重製必究！】

else A[k]=R[j]  
j=j+1

## (二)貪婪演算法

是一種在每一步選擇中都採取在當前狀態下最好或最佳的選擇，從而希望導致結果是最好或最佳的演算法：

### 1.步驟：

- (1)以數學模型來描述問題
- (2)將原始問題分成若干子問題
- (3)針對每一子問題求得最佳解
- (4)將子問題的最佳解合成原來問題的一個解，此解即是原始問題最佳解

### 2.例如：在一圖中求最小生成樹問題

- (1)將圖中的邊(edge)進行排序
- (2)從最小的邊選起，若欲選邊不會與已選邊產生迴圈，則將其加入MST中
- (3)掃描完所有邊後MST集合中的元素即組成最小生成樹(Kruskal演算法)

【版權所有，重製必究！】