

《資料處理》

一、請比較動態隨機存取記憶體 (DRAM)、快取記憶體 (Cache)、暫存器 (Register)、硬碟 (Hard Disk) 等儲存單元或記憶體裝置的資料儲存容量大小及存取速度，並說明計算機儲存結構為何要如此設計的原因。(25分)

試題評析 此題為電腦硬體架構之基礎記憶體觀念，是電腦運作相當重要的一環，屬於學生應該把握之送分題。

答
 容量大小是愈快的愈小：小到大分別是 register→cache→DRAM→Hard Disk。
 速度快慢由快到慢依序是：Register→Cache→DRAM→Hard Disk。

如此設計是為了配合不同的電腦元件需求，將密集運算配上較快讀寫速度的記憶體，以分工加速系統運行。同時也能夠降低開銷，因為速度愈快的記憶體愈貴，一般存放長期檔案則不可能用 DRAM 來放，而使用價格較低廉的 Hard Disk 存放。

二、關聯代數 (Relational Algebra) 可用來描述資料庫中關聯資料表之間的操作方式。給定兩個關聯資料表 A 及 B，試將以下關聯代數運算式轉成以 SQL 語法描述之。(25分)

$$\text{Result} = \pi_{\text{AuthorName}}(\sigma_{\text{Title} = \text{'Big Data'}}(A \bowtie_{A.No = B.No} B))$$

試題評析 關聯代數以及 SQL 語法之應用，逐層分析並轉換成 SQL 即可。

答
 SELECT AuthorName FROM (SELECT * FROM A,B WHERE A.No=B.No) as x WHERE Title='Big Data'
 Result 為從 A 和 B 表中以同 No 合併之表格，且 Title 為 Big Data 之資料的 AuthorName。

三、網際網路的快速發展，促成了企業與個人用戶在網路上編成各種形式的多媒體電子內容於網際網路上發表、聯繫、交易等傳播模式，每天因此生成大量及複雜的大數據 (Big Data)。大數據已經超越傳統資料庫管理軟硬體的處理能力，因此需要發展新的硬體設備及軟體技術。有別於傳統資料庫處理對象的數據集，請說明至少四種大數據具備的特徵為何？(25分)

試題評析 此題考近年流行之大數據(Big data)之特性以及相關概念，屬於較簡單之觀念題。

答
 大數據的特徵有以下幾點：
 (一)數據量級夠大：從以往的 TB 級躍升到 PB 等級。
 (二)數據類型夠多：數據型式包含文字、圖像、影像等。
 (三)價值密度低：例如連續影片中，有價值的影像可能只有 1、2 秒。
 (四)時效性：必須具備處理速度快、即時輸入、即時處理丟棄等特性。

四、堆疊 (Stack) 和佇列 (Queue) 分別是用來記錄先進後出 (First In Last Out) 和先進先出 (Fist In First Out) 等特定的工作排程形式，但若今有一工作排程需按照已進入準備排程的最小工作編號執行 (不同工作的編號不會重複)，則可採用鏈結串列 (Linked List) 來操作。給定一個已建置好的鏈結串列資料結構，其中的每個節點 (Node) 含有三個欄位，第一個欄位為 JobName，存放準備執行之工作名稱，第二個欄位為 JobID，存放該工作之編號，第三個欄位為指標欄位 Next，指向鏈結的下一個節點。另有一指標 Head，指向鏈結串列的第一個節點，而最後一個節點的指標欄位 Next，指向 NULL 代表串列的結束。請寫出虛擬程式碼 (Pseudo Code)，讀取鏈結串列中具有最小編號的工作名稱，並將該存放節點從鏈結串列中刪除並釋放

被刪除節點所占用的記憶體空間。虛擬程式碼需附帶適當的註解，說明操作的程序及理由。
(25分)

試題評析

考驗同學對於Linked List相關操作以及演算法虛擬碼撰寫之熟練度，由於Linked List操作上容易有盲點，故需花時間細心檢查虛擬碼之正確性。

答

算法：遍歷兩次，第一次找出最小 ID，第二次把該 ID 對應之 JobName 取出並刪除節點

```

// 設定一個指標 now 用來遍歷
now = Head
// 設定一個變數 m 存放最小編號 (初始設成一個很大的數)
m = MAX_NUM
// 存放最後的 JobName 的變數 name
name = ""
// 若一個元素都沒有，則不做任何事
if(now == NULL) return NULL
// 開始尋找最小編號
while(now->next != NULL) {
    now = now->next
    m = min(m, now->JobID)
}

// 遍歷第二次，把該 JobName 取出並刪除節點
now = Head
// 先檢查是否是開頭
if(now->next->JobID == m) {
    name = now->next->JobName
    Head = now->next->next // 將 Head 指向第二個節點
} else {
    while(now->next != NULL && now->next->next != NULL) {
        now = now->next
        if(now->next->JobID == m) {
            name = now->next->JobName
            now->next = now->next->next
            break
        }
    }
}
return name

```

【版權所有，重製必究！】