

《資料處理》

一、請根據資料庫中的資料儲存架構，說明以下3種不同的資料庫，所採用的儲存架構以及該架構的優缺點。（每小題10分，共30分）

- (一)階層式資料庫
- (二)網狀式資料庫
- (三)關連式資料庫

試題評析	資料庫各種儲存架構，算是考題中偏老掉牙的議題了。考生們應有準備。
考點命中	高上《資料處理講義》第二回，唐箏編撰，頁9-12。

答：

(一)階層式 (hierarchical) 資料模型

1. 儲存架構

- (1) 樹狀 (tree) 結構。
- (2) 以指標 (pointer) 連接父節點與子節點。
- (3) 可描述一對一及一對多關係，無法描述多對多關係。
- (4) 父子關係必須事先設定。
- (5) 整合性限制：除樹根 (root) 外，每個節點恰有一個父節點。

2. 優點

- (1) 存取快速、有效率。
- (2) 非常適合階層性的資料。
- (3) 適合大量資料，且結構固定的應用。

3. 缺點

- (1) 無法表示多對多關係。
- (2) 資料關係必須事先設定，無法動態產生資料間的關係。
- (3) 插入刪除動作複雜 (必須考慮樹狀結構)。
- (4) 資料獨立性低 (父節點刪除時，子節點也被刪除)。

(二) 網路式 (network) 資料模型

1. 儲存架構

- (1) 格狀 (Lattice) 結構。
- (2) 以指標 (pointer) 連結各節點。
- (3) 可描述一對一、一對多、及多對多關係。
- (4) 節點間關係需事先設定。
- (5) 整合性限制：除樹根 (root) 外，每個子節點的所有父節點必須存在，其中樹根可能不只一個，父節點亦可能不只一個。

2. 優點

- (1) 存取快速、有效率。
- (2) 可描述階層式資料模型無法表達的多對多關係。
- (3) 可達到資料獨立性。

3. 缺點

- (1) 結構最為複雜。
- (2) 指標過多、難理解。
- (3) 資料關係必須事先設定，無法動態產生資料間的關係。

【版權所有，重製必究！】

(三) 關連式 (relational) 資料模型

1. 儲存架構

- (1) 採用關連 (relation, 或稱表格 table) 為主要資料結構，每個關連含有表頭 (heading) 與主體 (body) 兩

部分。

- (2) 關連式資料模型無指標的存在，關連間相同欄位以「外鍵 (foreign key)」相連結。
- (3) 關連必無重複的紀錄 (record, tuple, row)，且紀錄 (record, tuple, row) 的順序不重要，屬性 (attribute, field, column) 的順序亦不重要。
- (4) 整合性限制：必須服從「個體整合性限制 (Entity integrity constraints)」與「參考整合性限制 (Referential integrity constraints)」。
- (5) 為企業最常使用的資料模型。

2. 優點

- (1) 理論簡單、易於實行。
- (2) 接近使用者角度。
- (3) 可達到資料獨立性。
- (4) 查詢方式簡易，採宣告式 (declarative) 查詢語言。
- (5) 最為普遍使用。

3. 缺點

- (1) 效率較階層式、網路式差。
- (2) 不適用大量資料的處理。

二、Ethernet (乙太網路) 是一種廣泛使用的電腦區域網路技術。

(一) 請說明早期的Ethernet的主要連線架構。(5分)

(二) Ethernet採用CSMA/CD來解決多部電腦間，資料傳輸相衝突的問題，請說明CSMA/CD的工作原理。(20分)

試題評析	乙太網路是區域網路中使用最普遍的架構。今年資料處理的前兩題較偏基本題型，希望考生這兩題可取得較高分數。
考點命中	高上《資料處理講義》第一回，唐箏編撰，頁63-64。

答：

(一) Ethernet：Ethernet採用匯流排 (Bus) 區域網路架構，將所有電腦或終端機連接至一條共同線路 (匯流排)，透過該共同線路貫穿整個網路。優點是架構簡單、成本低，流量少時效率高，擴充容易，電腦間可分工。但缺點是流量大時碰撞嚴重效率急劇降低，匯流排損毀則網路失敗。

(二) CSMA/CD運作方式：

- (1) 訊號採用廣播方式傳送，因此可能發生碰撞。
- (2) 當節點要發送訊號時，會先偵測通道是否有其他節點正在使用 (carrier sense)。
- (3) 當通道沒有被其他節點使用時，就傳送封包。
- (4) 封包傳送之後立即檢查是否發生碰撞 (carrier detection)，若是發生碰撞則對通道發出高頻訊號告知其他節點已經發生碰撞。
- (5) 碰撞後隨機等待一段時間重新發送封包。
- (6) 嘗試15次都失敗的話則告知上層Timeout。

三、在作業系統領域，有所謂的分時 (time-sharing) 作業系統。

(一) 請說明其工作原理。(5分)

(二) 請說明外部中斷 (Interrupt) 如何協助作業系統提供分時 (time-sharing) 功能。(5分)

(三) 當中央處理機 (CPU) 無法接受外部中斷時，小明突發奇想，提出利用內部中斷 (Trap) 指令的替代方案。小明宣稱只要將Trap指令，適當的插入機器程式碼中，並適當的修改作業系統，仍然可以讓作業系統達到高度的分時 (time-sharing) 功能。請對小明的奇想，提出評論。你必須先回答，同意或不同意小明的論點 (請勿提供模稜兩可的答案)，再說明你同意或不同意的理由。(10分)

試題評析	第三題是作業系統的題型。分時作業是OS中很常用到的技術；但第三小題稍偏應用，同學們需要對內部、外部中斷觀念清楚，思考作答。
考點命中	高上《資料處理講義》第一回，唐箏編撰，頁16。

答：

- (一)Time-Sharing (分時作業)：Multiprogramming的一種，允許系統內存在多個process同時執行，透過CPU scheduling技術，各程式分配一段時間輪流交替執行，可提高CPU使用率(utilization)。電腦Memory內有2個以上互不相關的程式可同時被執行，CPU交替執行之，使得User產生電腦專屬執行某一程式的錯覺。
- (二)外部中斷，指CPU以外的週邊元件所發出的中斷，如：I/O Complete, I/O Error, machine check等。外部中斷可協助time-sharing的scheduling，例如：採用Round-Robin方式，規定CPU的執行時間區間，若process取得CPU後，未能在quantum內完成工作，則由外部元件執行外部中斷，強迫process放棄CPU，讓下一個process執行。
- (三)不同意。內部中斷(Trap)指令，必須插在程式碼中，即程式必須執行到該處，始可進行中斷。即每隻程式皆必須插入該程式碼。但若程式執行特定操作時間過長，導致程式碼無法向下執行至指定處，則無法進行中斷，也無法達成分時的效果。

四、為簡化程式設計細節，本題假設，對程式所將輸入的數字，都會是大於2的正整數。

- (一)小明設計了一個演算法，用來判斷所輸入的一個正整數，是否為質數。小明的演算法，用虛擬碼寫出如下：

```

procedure check_prime(N) {
  isprime = TRUE
  for (i = 2; i <= N; i=i+1) {
    If (N 是i的倍數) then isprime=FALSE
  }
  print(isprime)
}

```

小華將小明的演算法輸入電腦，發現演算法有錯誤。小華跟小明詢問，發現小明的觀念沒錯，只是打字時不小心。請問，小華該如何為小明改正這個演算法？(註：你只能更正演算法中打錯誤的部分，不可以重新設計新的演算法)(10分)

- (二)小華是個遞迴演算法的愛好者。所以，小華嘗試用遞迴程式呼叫，重新設計了一個，也能判斷所輸入的一個正整數，是否為質數的遞迴演算法，交給小明。結果小明不小心弄丟了。現在，請你為小明重新設計這個，能用來判斷所輸入的正整數是否為質數的遞迴演算法，請注意，在你的演算法中不可以使用任何的迴圈。(15分)

試題評析	今年的資料處理，不考程式，僅需寫演算法。至少在程式語法上較不需太過精雕細琢。但是這兩小題都是屬於應用題，第一小題把題目演算法讀懂，應該不難取分。第二小題解法不見得是唯一的，考生們需要先做質數運算的基本邏輯思考，再將其轉為遞迴演算法。
考點命中	高上《資料處理講義》第三回，唐箏編撰，第七章。 高上《資料處理總複習講義》，唐箏編撰，頁36。

答：

- (一)for迴圈應去除i<=N的等於，即修改為：

```
for (i=2; i<N; i=i+1) {
```

因為原演算法for迴圈最後一次會結束於i=N，但如此N必為i的倍數(一倍)，故找不出質數。質數判斷時，最多僅能除到i=N-1，才可正確找出質數。

- (二)

```
main (int N) {
```

```
check_prime(N, N/2+1)
}

procedurecheck_prime(N, k) {
  if(k==1) then return true; //從N/2+1一路除到1都找不到可整除的數字，表示其為質數
  if (N%k==0) then return false; //如果可整除，表示N非質數
  else return check_prime(N,k-1); //若非上述，表示尚未找到可整除者，繼續將除式減一往下找找看
}
```

高
點
·
高
上

【版權所有，重製必究！】