

《資料處理》

| | |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 試題評析 | 今年考題，在計算機基礎概要的部分，分數比重較以往為重，共佔40分；資料結構章節卻反常的一題都沒有；網路和資料庫，仍為資料處理之重點，各佔了40分與20分。 資料結構未出現在考題中，相信對非資訊背景的同學而言，應可小小鬆了一口氣；今年考題，沒有出現過深或刁鑽的考題，反之，在考驗同學，對一些資料處理底層技術，是否涉獵夠廣夠細。若有對計算機、網路、資料庫三科多下功夫的同學，尤其是傳統計算機及網路技術概念，今年應可得到一定水準的成績。但若將重點放在資料結構、程式語言的同學們，反而會有些小小失望。 |
| 高分命中 | 第一題：唐筭《資料處理講義第一回》頁2-3、6，唐筭《資料處理總複習講義》頁16。 第二題：唐筭《資料處理講義第三回》頁5-6、13-16、19。 第三題：唐筭《資料處理講義第一回》頁44-46。 第四題：唐筭《資料處理講義第一回》頁55。 第五題：唐筭《資料處理講義第一回》頁13-14 |

一、電腦系統中，資料可能存在於階層式記憶體系統的各個層次中，其中CPU可以直接存取的記憶體有那些？CPU執行時，是透過那兩個主要單元在運作？在CPU與輸出入（I/O）設備的溝通中，可以藉由那些技術的應用，以協調完成I/O的動作？請詳細說明其運作。（20分）

答：

- (一)一般CPU中，包含暫存器與快取記憶體兩類記憶體系統。暫存器的速度最快，但價格昂貴、儲存空間小；快取記憶體其次。快取記憶體常見分為L1與L2快取記憶體，L1快取多半位於處理器晶片中，執行速度較快，L2快取通常是CPU模組的一部分，容量比L1稍大，速度也稍慢。此外，CPU也可存取CPU外的主記憶體(Main Memory)做資料處理。
- (二)CPU執行時，透過控制單元與算術及邏輯運算單元運作。控制單元(Control Unit, CU)控制計算機的所有作業程序以及所有其他單元間動作的協調與資料的傳遞；算術及邏輯運算單元(Arithmetic Logic Unit, ALU)執行如加減乘除的算術運算以及邏輯上的分析、比較、判斷、轉移等運算。
- (三)CPU的溝通是透過匯流排(Bus)來傳輸資料。匯流排是CPU、記憶體與I/O裝置訊號傳遞的共同通道。一次只能有一個裝置去使用匯流排，所以當同時有兩個以上的裝置使用時，必須由主機板上的I/O控制電路加以協調，先將優先權較高的裝置使用；等該裝置使用完畢後，才輪到其他裝置使用。匯流排的分類：
 - 1.位址匯流排(Address Bus)：負責傳送CPU所要存取資料的位址，它可以決定CPU所能處理的記憶體容量。
 - 2.資料匯流排(Data Bus)：負責傳送CPU所要存取的資料，其線數的多少代表CPU的字組Word，亦即CPU一次所能存取資料的基本單位。
 - 3.控制匯流排(Control Bus)：負責傳送CPU所發出的控制訊號。

二、關聯式資料庫的設計中，多個交易的處理，依序的序列排程(serial schedule)可以保證資料的一致性。然而，不一定各個交易都要依序執行，只要是合法排程序列(serializable schedule)，也可以獲得資料的一致性。請解釋何謂合法排程序列？在交易處理的各個命令操作中，存在有關連性之先後執行次序關係。並行處理時，需要考慮這些命令的關連性，以確保交易間不會相互干擾，也就是所謂的區隔性(isolation)。我們一般會以鎖定(lock)與時戳(timestamp)的機制來實作區隔性，請說明資料讀取(read)時，時戳機制下是如何進行的？鎖定機制下又是如何運作的？（20分）

答：

- (一)合法排程序列(serializable schedule)：一個n個交易的排程是合法(serializable)的，假若此排程與相同的n個交易的某個序列排程(serial schedule)等價(equivalent)。而等價又可分為衝突等價、景觀等價等不同技術。以衝突等價為例，若兩個排程中，發生衝突的順序相同，稱此兩排程為衝突等價。兩交易操作間，只要針對同一項目有任何寫入操作，則此兩操作為發生衝突。【高雄】高雄市新興區中山一路308號8樓·07-2358996
- (二)時戳機制(timestamp)：每項交易賦予一個時戳，用以判斷該交易進入系統的時間順序。交易T若欲讀取某

資料項目時，若比此交易晚進入系統之交易已寫過此資料項目，則交易T操作不允許，交易T將被取消與回復；反之，則可執行Read操作，並將讀取時戳設為原本時戳與交易T中較大者。

- (三)鎖定(lock)：鎖定(locking)是一個與資料庫中資料項目結合的變數，描述了資料項目的狀況，決定何種動作允許或不允許應用在此資料項目上。鎖定技術主要為「二元鎖定(Binary locking)」、「共享互斥鎖定(Shared and Exclusive locking)」與「兩階段鎖定(Two phase locking)」。基本觀念為交易讀取時，若該資料項目已被其他交易佔用(鎖定)，則此交易不允許讀取。上述不同技術描述了讀取操作的不同並行程度。例如：共享互斥鎖定中，若資料項目被其他交易「讀取鎖定」，則交易仍可再次讀取，但若被其他交易「寫入鎖定」則不允許讀取。

三、電腦網路的通訊協定，常會以層次的概念來設計，請問此種方式的設計有何優點？在資料傳輸的過程中，上下層資料會有所不同，請問這兩種資料在接收與傳送時是如何完成的？在TCP/IP網路中，我們常會以IP位址來表示目的地端的位置，由於協定底層的傳輸，並無法藉由IP位址找出目的地端電腦的實際位置。請詳述傳送端的電腦是如何將資料送往此IP位址對應的目的地端電腦。(20分)

答：

(一)網路層次模型的優點：

- 1.定義明確：訂定共同的術語來描述網路的功能，將可對運作有更深入了解且可更順利的協同運作。
- 2.分工明確：分割程序，讓每一項技術只執行一項功能，以使各項技術可以分別發展。例如：無線媒體技術的發展，不是因路由器的發展才跟著進步。
- 3.開放架構：促進競爭，因為不同供應商的產品都可以協同運作，沒有任何一種產品是獨一無二。
- 4.共同標準：提供了一個共同的語言，來描述網路的功能和機制。

輔助協定的設計，因為特定階層的協定會定義其本身的規範，以及與上層和下層之間的介面

(二)傳送時：每一層依照其負責功能，加上封包標頭(header)後，傳送至下層；

接收時：每一層將其層標頭(header)解開後，再傳送至上層。

(三)當封包要在實體網路傳送時，除了IP位址之外，還必須知道通訊雙方的主機實體位址(Physical Address/MAC Address)才能將資料確實傳送到實體主機中。使用ARP或RARP處理「IP位址對應實體位址」及「實體位址對應IP位址」的位址轉換工作。「位址解譯協定」(Address Resolution Protocol) ARP：是負責將IP位址轉換成Mac實體位址的一種通訊協定，當某一台電腦要傳送資料到某個IP位址時，會先傳送ARP封包詢問網路上哪台電腦的MAC Address對應到這個IP位址，當目的端的電腦接收到這個ARP封包之後便會回應給來源電腦進行資料傳送。

四、CSMA/CD與CSMA/CA是電腦網路上普通被使用的通訊協定，請問兩者在使用環境及特性上有何差異？其中CD與CA分別代表什麼？又如何達成？(20分)

答：

(一)CSMA/CD(Carrie Sense Multiple Access with Collision Detection)多應用在有線網路上，例如：乙太網路(Ethernet)的標準IEEE 802.3；而CSMA/CA(CSMA/CA, Carrie Sense Multiple Access with Collision Avoidance)則常用於無線網路應用，例如：IEEE 802.11。

(二)CD：Collision Detection，碰撞偵測。CA：Collision Avoidance，碰撞避免。

(三)CSMA/CD運作方式：

- 訊號採用廣播方式傳送，因此可能發生碰撞。
- 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)。
- 當通道沒有被其他節點使用時，就傳送封包。
- 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。
- 碰撞後隨機等待一段時間重新發送封包。
- 嘗試15次都失敗的話則告知上層，Timeout。

CSMA/CA運作方式

- 訊號採用廣播的方式傳送(非常容易受到無線電波干擾)

- 當節點要發送訊號時偵測頻道是否空閒。
- 若是空閒則等待IFS(Interval Frame Space)隨機時間後再次偵測頻道是否空閒。
- 若是空閒則發送封包，反之重新進入等待頻道空閒(隨機等待時間)。
- 發送RTS之後必須在限定時間內收到來自目的端的CTS訊號。
- 當失敗32次之後通知上層Timeout。

五、在一般的認知裡，高階程式語言與低階程式語言在可攜性上有很大的差別。請問高階語言之所以較具可攜性的原因為何？將一個高階程式的原始碼轉換成目的碼，需要經過那些過程？從目的碼到可以執行的執行碼又需要經過那些步驟？這些過程或步驟的作用為何？（20分）

答：

- (一)越高階的程式語言，越容易被人類閱讀，且與硬體無關，因此可攜性高；但低階語言，如：機器語言、組合語言，皆必需符合特定機器或CPU，可攜性較低甚至不具備可攜性。
- (二)程式碼從原始碼轉為可執行碼，依據程式可分為直譯(interpreter)與編譯(compiler)兩種。
- 1.直譯(interpreter)：直接將原始碼轉換至可執行碼。
 - 2.編譯(compiler)：先將原始碼轉為目的碼，再將目的碼轉換至可執行碼。
- 高階程式原始碼透過編譯器轉換為目的碼，需經過以下步驟：
- 1.字義分析、2.語法分析、3.最佳化。
- 目的碼轉換至可執行的執行碼，則一般需透過載入程式(loader)經過以下步驟：
- 1.連結(linking)、2.重新定址(relocation)、3.載入(loading)。

高上高普特考

高點·高上高普特考 goldensun.get.com.tw 台北市開封街一段2號8樓 02-23318268

【中壢】中壢市中山路100號14樓·03-4256899

【台中】台中市東區復興路四段231-3號1樓·04-22298699

【台南】台南市中西區中山路147號3樓之1·06-2235868

【高雄】高雄市新興區中山一路308號8樓·07-2358996

【另有板橋、淡水、三峽、林口、羅東、逢甲、東海、中技、雲林、彰化、嘉義】