輸出輸入系統

資科系
林偉川

I/O 硬體

- 裝置與電腦的溝通是透過連接點或埠(port)
- 埠為對應到相對裝置的記憶體位址或硬體指令索引，是軟硬體溝通的直接管道
- 多種裝置使用共同的一組導線，則此導線為匯流排
- 現今的電腦系統有許多週邊裝置，如
  - 儲存裝置：磁碟、軟碟、光碟
  - 傳輸裝置：網路卡、數據機
  - 人機操作介面：螢幕、鍵盤、滑鼠

2

I/O 硬體

- PCI 匯流排將較快的裝置連接到處理器與記憶體等相關子系統，如
  - 圖形顯示控制器
  - 記憶體控制器
  - IDE 磁碟控制器
- 延伸匯流排連接速度較慢的裝置，如
  - 鍵盤
  - 串列埠(其控制器可以是單一晶片或是晶片的一部份)
  - 平行埠

3

I/O 硬體

- SCSI匯流排控制器通常獨立做成一塊電路板或主機介面卡，其中包含了處理器、微碼和記憶體，因此能處理更複雜的傳輸協定
- 內建控制器：例如磁碟內的磁碟控制器，負責的工作如錯誤磁區對映(bad-sector mapping)、預先讀取(prefetching)、緩衝(buffering)及快取(caching)
- 處理器與控制器共同架構命令與資料在匯流排上傳遞的方式

4

典型的PC匯流排結構 I/O硬體

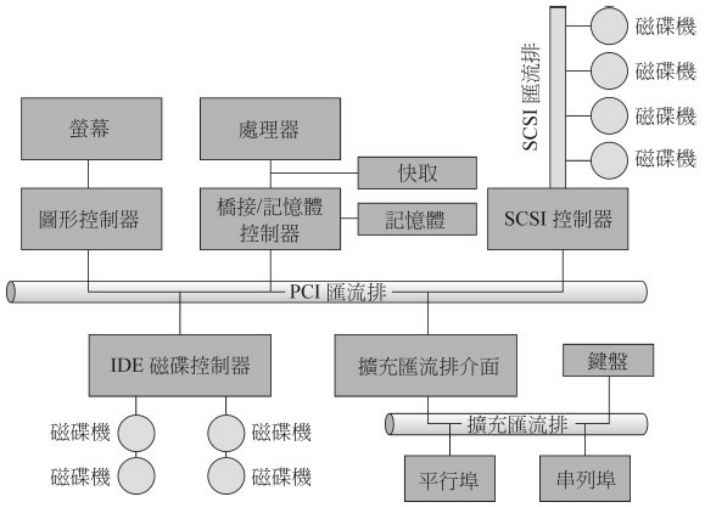
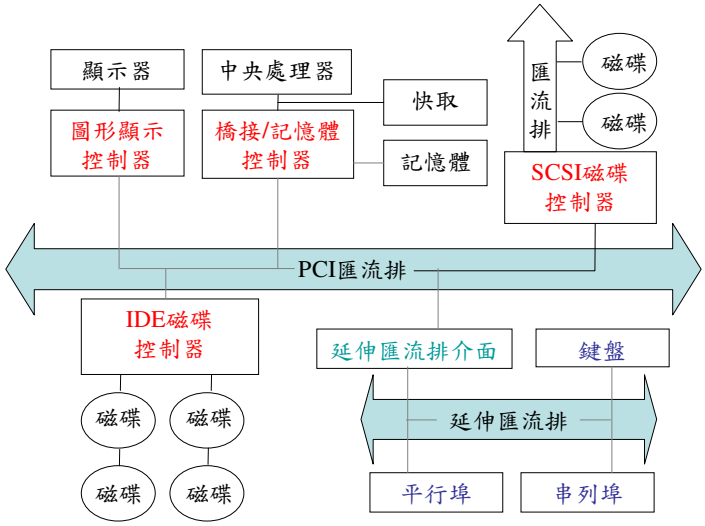


圖 13.1 典型的 PC 匯流排架構。

5

典型的個人電腦匯流排結構



6

I/O 硬體

- 控制器有許多處理資料和控制訊號的暫存器，而處理器藉由讀取與寫入這些暫存器，來與處理器相互傳遞訊息
- 處理器與控制器利用一些特殊的I/O指令傳輸一個位元或字元到I/O埠的裝置暫存器上，並觸發匯流排選擇適當的裝置來傳遞或取出裝置暫存器上的位元(I/O對映I/O)

7

I/O 硬體

- 另一種方法為裝置控制器支援記憶體對映I/O，將裝置控制暫存器對映到處理器中的一個位址空間，當CPU執行I/O要求時，只要對該位址空間執行記憶體的存取指令，即可讀取或寫入裝置控制器內的暫存器(記憶體對映I/O)
- PC的I/O裝置埠位址如下：

8

I/O位址	裝置
000-00F	DMA控制器
020-021	中斷控制器
040-043	計時器
200-20F	遊戲控制器
2F0-2FF	次要串列埠
320-32F	磁碟控制器
378-37F	平行埠
3D0-3DF	圖形顯示控制器
3F0-3F7	軟碟控制器
3F8-3FF	主要串列埠

9

I/O 硬體

- 典型I/O埠包含4種暫存器，分別為狀態、控制、資料輸入及資料輸出暫存器
- 狀態暫存器表示目前的狀態➔處理命令是否已經執行完畢
- 控制暫存器可觸發一個命令或改變裝置的模式，串列埠內有1個位元來選定以全雙工或半雙工做為通訊模式，1個位元來進行同位檢查，1個位元可設定字元長度為7或8個位元，其他位元則用來設定此串列埠的傳輸速度

10

I/O 硬體

- 資料暫存器是由1-4個位元組所組成，有些控制器有先進先出(FIFO)晶片負責處理多位元組的輸入或輸出(AX,BX,CX,DX)
- 延伸控制器則處理超出資料暫存器容量的資料，一般經由FIFO晶片暫存少量突發的資料，直到裝置或主機能夠接受這些資料為止

11

輪詢(Polling)

- 主機和控制器間簡單的協調機制：控制器和主機上用2個位元作為協調的工具，其中控制器狀態由狀態暫存器中的忙碌位元表示，1表示忙碌中，0表示空閒
- 主機要求藉由命令暫存器的命令就序位元來掌控，1表示就緒，0表示未完成

12

輪詢(polling)

- 握手協定(handshaking)
  - 假設有2個位元用來協調控制器主機之間的生產-消費關係，控制器透過設定在狀態暫存器內的忙碌(busy)位元表示本身狀態[回想設定位元表示將此位元值設為1，而清除位元表示將此位元值設為0]，當控制器忙於工作時，即設定忙碌位元，而當準備好可以接收下一個指令時，即清除忙碌位元。

13

輪詢(Polling)

- 主機要輸出資料到某一埠時，控制器針對每一個要傳送的位元組，執行的協調程序如下：
  - 主機重複讀取忙碌位元，直到此位元變為0
  - 主機設定命令暫存器內的寫入位元，並將一個位元組寫入資料輸出暫存器
  - 主機設定命令就序位元
  - 當控制器察覺到命令就序位元被設為1，則設定忙碌位元，表示有命令正在處理

14

輪詢(Polling)

- 控制器讀取命令暫存器並查看所寫入的命令後，得知要讀取資料輸出暫存器內的資料，並且執行裝置I/O
- 控制器清除命令就序位元及狀態暫存器內的錯誤位元，以表示裝置成功地執行I/O，並清除忙碌位元，表示可處理下一個要求

- 主機一遍又一遍地讀取控制器狀態暫存器內的值，並處於忙碌等待狀態，這種I/O的方式稱為輪詢
- 如果控制器和裝置速度相當快，這種方法非常實用

15

輪詢(Polling)

- 但是在較慢的裝置上則會因為等待太久，而浪費CPU的資源
- 若資料來源是串流(streaming)時，若主機不能快速處理，在擁有小緩衝區的控制器上可能會發生溢位，而導致資料流失的問題
- 現今CPU速度只要3個指令週期就足夠輪詢一個裝置，所以可以滿足基本輸出入的要求，且在需要重複執行I/O的情形下，輪詢會十分有效率

16

輪詢(Polling)

- 輪詢不適用於
  - 大量存取資料
  - 速度緩慢的控制器

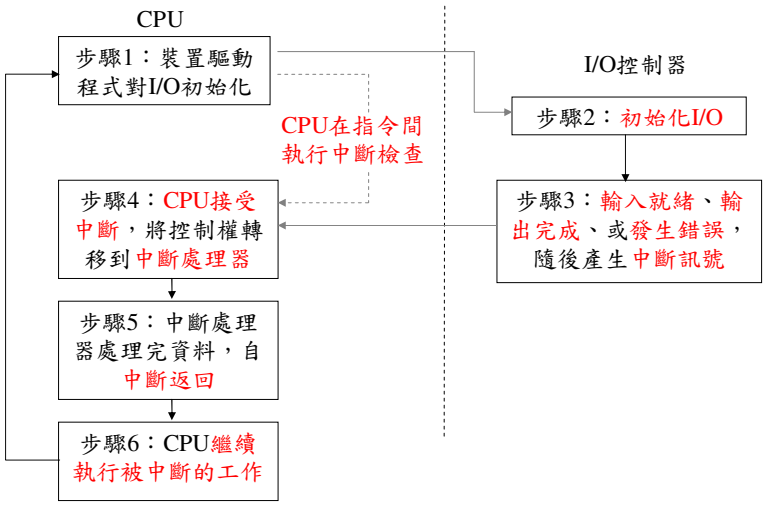
17

中斷

- 中斷是可以改變 CPU 待執行指令執行順序的事件
  - CPU 有個硬體機制稱為中斷要求線
  - 當有中斷發生(來自控制器)，CPU 會停止目前執行的指令，並執行中斷處理常式
- 中斷處理常式
  - 中斷發生後 CPU 處理中斷需求的程式
- 中斷向量
  - 不同的中段需求利用中斷向量，對應到中斷處理器相對的不同中斷處理常式

18

中斷驅動I/O程序



19

中斷

- 複雜的中斷處理的功能要提供：
  - 在關鍵的處理程序內，延緩不重要的中斷處理
  - 有效的方法配置適當的裝置給中斷處理器，而不需要事先輪詢所有裝置，以查看是哪一個裝置產生中斷
  - 提供巢狀式中斷，讓中斷處理時，還允許被中斷
- CPU提供可遮罩與不可遮罩中斷
  - 可遮罩中斷，中斷訊號送至處理器的 INTR 腳位，它們能夠藉由清除 eflags 暫存器上的 IF 旗標而被暫時關閉(裝置控制器都是此類的中斷)
  - 不可遮罩中斷，訊號送至處理器的 NMI 腳位，無法經由清除 IF 旗標將它們暫時關閉(硬體錯誤)

20

中斷

- 中斷向量內存著各個中斷處理器相對的記憶體位址，中斷處理機制所接受的參數，則代表在中斷向量內的索引
- 一般使用中斷串列，也就是在中斷向量表內每個紀錄都指向中斷處理器所組成的串列
- 當有中斷發生時，會逐一檢查對應串列內的處理常式，直至找到相映的中斷處理常式為止，如此可避免使用大型的中斷向量表所造成的額外負擔
- 以Intel Pentium CPU的中斷向量的設計，0-31的中斷屬於NMI，用來驅動不同的錯誤處理；32-255的中斷則屬於MI，用來處理由各種裝置引發的中斷

21

向量索引	描述
0	除法錯誤
1	偵測例外中斷
2	NMI中斷
3	暫停卓
4	偵測溢位
5	邊界範圍例外中斷
6	無效運算碼
7	裝置未就緒
8	雙重(分頁)錯誤
9	協同處理器分段溢位覆蓋
10	無效工作狀態分段

22

向量索引	描述
11	分段不存在
12	堆疊錯誤
13	一般保護
14	分頁錯誤
15	未使用
16	浮點數錯誤(floating-point error)
17	對位檢查(Alignment check)
18	機器檢查
19-31	未使用
32-255	可遮罩中斷

23

中斷

- 中斷也有優先權等級機制，使CPU可以延緩低優先權中斷的處理，讓高優先權的中斷搶先低優先權中斷處理的執行
- OS啟動時會偵測硬體匯流排，以判斷哪些裝置在線上，並設置相對應的中斷處理常式到中斷向量表

24

中斷

- 另一種中斷實作的方式稱為系統呼叫
  - 提供應用程式呼叫核心服務常式，系統呼叫會檢查應用程式給定的參數、建立資料結構、傳遞資料到核心，並執行特殊指令，系統呼叫執行時，中斷處理的硬體會儲存使用者程式目前的狀態、並切換至核心模式，執行所要求的處理程序
- 軟體中斷
  - 可由撰寫程式來觸發的中斷，被賦予較低的中斷權限

25

中斷

- 陷阱
  - 軟體觸發、事先設計好的中斷
- 中斷也可管理核心內的控制流程，例如硬碟讀取，其中步驟是從核心空間拷貝資料到使用者的緩衝區內，這種複製的動作並非十分迫切，所以被設定成低優先權的中斷

26

中斷

- 要提高硬碟的使用率，應儘可能在完成前一指令後，立即執行下一個I/O，所以將執行下一個I/O設定為擁有較高的優先權。利用中斷處理器時做一個硬碟讀取程式時，系統會讓高優先權的中斷先紀錄I/O狀態、清除裝置中斷、執行下一個迫近的I/O、再產生一個低優先權的中斷，來完成剩餘的硬碟讀取工作

27

直接記憶體存取

- CPU查看裝置狀態，一次傳送1個位元組資料給控制暫存器的I/O程序，程式化I/O(PIO)
- 避免PIO加重 CPU 的負擔，將記憶體存取的工作交由直接記憶體存取控制器(DMA)來處理，此時 CPU 便可繼續其他非 I/O 的指令
- 系統將DMA指令寫入記憶體內，指令包括傳輸來源指標、目的指標和傳輸的資料量，CPU將指令位址寫至DMA控制器後，便可繼續執行其他工作

28

直接記憶體存取

- DMA控制器則進行記憶體匯流排的操作、放置相關位址至匯流排進行傳輸，這些動作都不需要CPU幫助
- DMA控制器和裝置控制器間的協調，藉由DMA要求線及確認線所完成。當一個位元組準備要傳輸時，裝置控制器會放置一個訊號到DMA要求線上，此訊號使得DMA控制器獲得記憶體匯流排的控制權，並放置目的位址至記憶體位址線上，之後傳送DMA確認訊號至DMA確認線上

29

直接記憶體存取

- 當裝置控制器接受DMA確認訊號後，就將該位元組傳送至記憶體內，並移除DMA的要求訊號。整個傳輸完成後，DMA控制器會發出中斷，告訴CPU傳輸已完成
- 當DMA控制器使用記憶體匯流排時，CPU對主記憶體的存取將被阻隔，此時CPU只能對快取內的資料進行存取，所以使用DMA雖然會降低CPU的計算速度，但是將資料傳輸工作交由DMA控制器處理，仍能提升系統整體效能

30

直接記憶體存取

- 具有記憶體保護下的核心會預防行程直接對裝置作處理，以避免資料被不當存取也可避免因裝置不當使用，而導致系統癱瘓
- 若核心不具保護模式，行程能直接存取裝置控制器這樣可以避免核心的協調、內文切換和多層的軟體階層所造成的延遲，而比較有效率。但是如此會降低系統的安全性及穩定性

31

直接記憶體存取

- 一般OS接提供具有記憶體保護和裝置的模式，使系統能對錯誤或惡意的行為加以防範及保護
- 直接記憶體存取的特性
  - 不透過 CPU，一次存取大量資料
  - 減少中斷次數
  - 適用於高速 I/O 設備

32

直接記憶體存取(DMA)

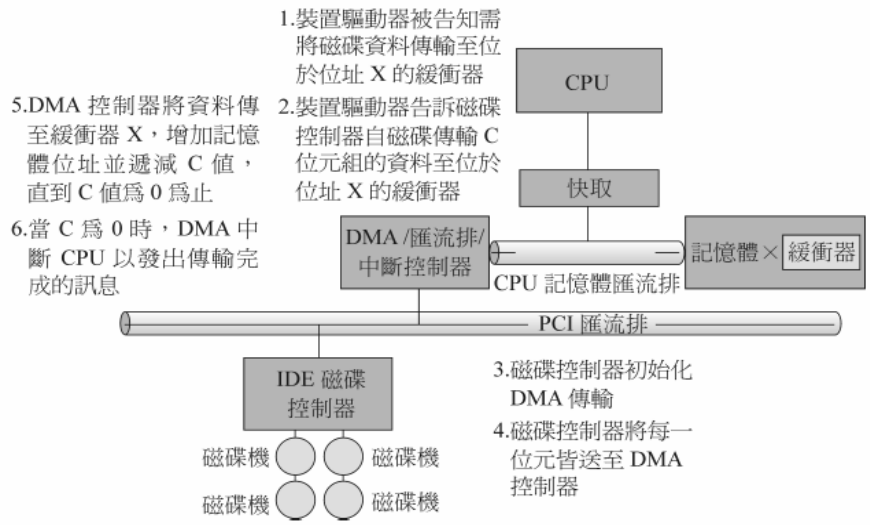
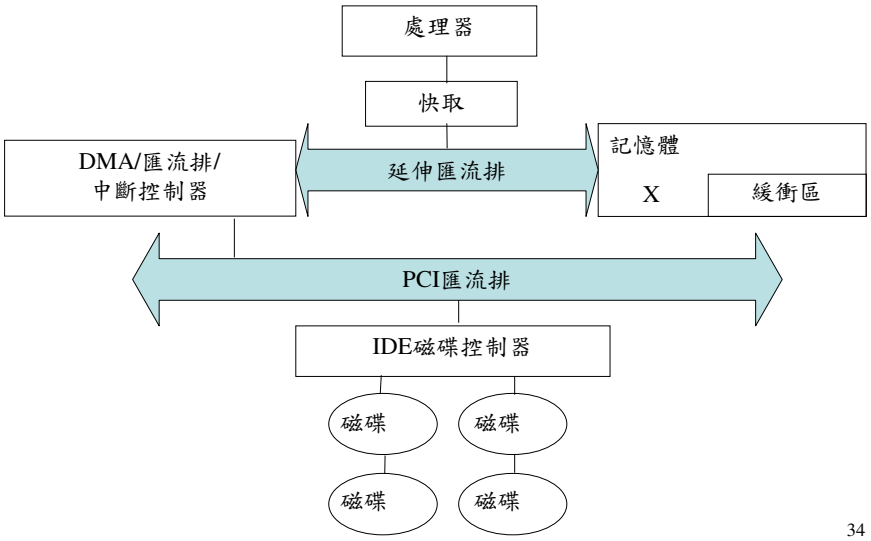


圖 13.5 DMA 傳輸之步驟

DMA傳輸步驟



34

DMA傳輸步驟

1. CPU告知裝置驅動程式傳送部份磁碟資料到位址X的緩衝區
2. 裝置驅動程式告訴磁碟控制器從磁碟上傳送C個位元組到位址X的緩衝區
3. 磁碟控制器初始化DMA傳輸
4. 磁碟控制器傳送每個位元組到DMA控制器
5. DMA控制器傳送該位元組到緩衝區X，遞增X並遞減C
6. 當C為0，DMA中斷CPU表示傳輸完成

35

應用系統I/O介面

- 使OS標準化地對待所有I/O裝置，應用程式開檔卻不需知道磁碟種類及新裝置加入電腦系統，不會影響OS正常運作 (PnP、UPnP)
- 抽象化、封裝化和軟體階層化，更重要是找出I/O的一般特徵，將I/O裝置間的差異性抽離出來，由一組標準的介面加以存取，裝置間的差異則被封裝在裝置驅動程式
- 裝置驅動程式對外採用標準的溝通介面，而內部則根據不同的裝置而量身訂做

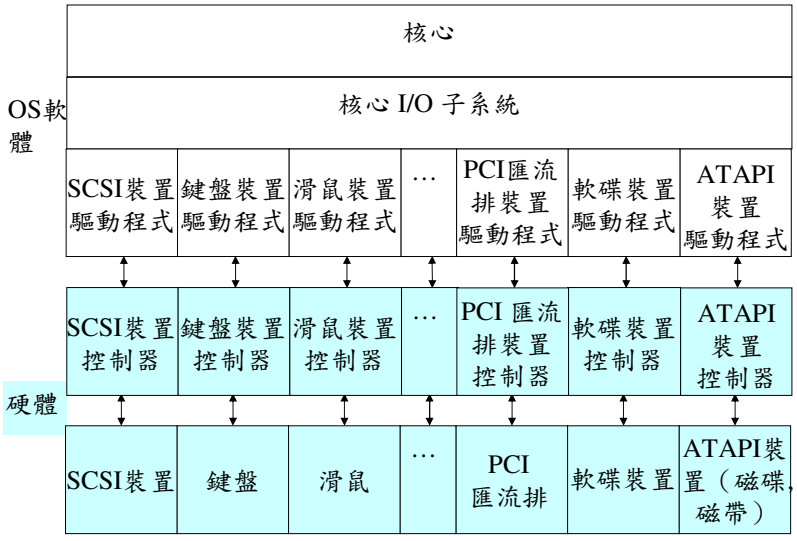
36

應用系統 I/O 介面

- 裝置驅動器階層化(軟體階層化)
  - 目的是隱藏裝置間的差異，使 I/O 子系統能獨立於硬體之外，簡化OS開發者的工作，硬體製造商也能夠設計新的且相容於現存主機控制器介面的硬體裝置，並撰寫新的裝置驅動程式到現行的OS，使新的周邊裝置不需要等待OS廠商支援即可使用
  - 一個特定裝置需要多種版本的驅動程式

37

核心 I/O 結構



38

應用系統 I/O 介面

- 不同裝置的變化
  - 字元串流或區塊：字元串流以一個位元組接著一個位元組的方式來傳輸，而區塊裝置的傳輸單位是多個位元組區塊為單位
  - 循序或隨機存取：循序存取裝置以裝置所決定的固定順序存取資料。隨機存取裝置可由使用者搜尋任何位置的有效儲存資料
  - 同步或非同步：同步裝置在傳輸期間，系統與裝置之間持續溝通，其反應時間可以預測。非同步裝置是建立連結後，系統與裝置是資料傳輸情形不定期溝通，其反應時間不規則，也無法預測

39

應用系統 I/O 介面

- 共享或專屬：共享裝置能同時提供多個行程或執行緒使用。專屬裝置則只提供專屬的行程或執行緒使用
- 操作的速度：從B/S到GB/S不等
- I/O方式：有些可以同時執行輸入及輸出，有些只提供單一方式的輸入或輸出

40

資料傳輸模式	字元串流 區塊	終端機 磁碟機
存取方式	循序 隨機	數據機 磁碟機
傳輸模式	同步 非同步	磁帶機 鍵盤
共享	專屬 可共享	印表機 記憶體
裝置速度	機器延遲 搜尋時間 傳輸率 操作間延遲	磁區轉到磁頭 磁頭移到磁柱 磁碟資料傳輸 連續兩次鍵盤輸入
I/O方式	唯讀 唯寫 讀寫	光碟機 圖形控制器 磁碟機

41

應用系統 I/O 介面

- 裝置存取方式分為 4 大類
  - 字元串流 I/O
  - 區塊 I/O
  - 記憶體對映檔案存取
  - 網路插口 (socket ➔ packet)
- 大部分的OS也提供一種後門式的系統呼叫，以函式指標的形式，將任何指令交給裝置驅動程式執行(UNIX之ioctl()三個參數)

42

區塊和字元裝置

- 區塊裝置介面泛指存取磁碟及其他區塊裝置的所有功能包括讀寫裝置的read()、write()命令，判斷是否為隨機存取的裝置，和可以搜尋到下一個傳輸區塊位址的seek()命令
  - 應用程式將區塊裝置當作一個線性區塊陣列使用裸I/O(raw I/O)
  - 應用程式不需要知道裝置在低階處理時的差異
  - 讀取、寫入和搜尋是存取區塊裝置的基本操作介面，使得應用程式不需要知道裝置在低階處理時的差距

43

區塊和字元裝置

- 記憶體對應檔的存取式安排在區塊裝置驅動程式的最上層，不必提供讀取操作，記憶體對應介面可透過主記憶體中的位元組陣列存取，直接對磁碟提供存取功能
- 將檔案對應至記憶體的系統呼叫，只會回傳包含複製檔案之位元組陣列的虛擬記憶體位址，至於實際的資料傳輸，只有在需要存取記憶體時才會被執行，這樣的傳輸處理與用來處理需求分頁的虛擬記憶體存取使用相同的處理機制

44

區塊和字元裝置

- 記憶體對應對程式設計師而言相當方便，存取記憶體對應檔就如同一般記憶體的讀取和寫入，因此採用虛擬記憶體的OS，大都以記憶體對應介面提供核心服務，如存取磁碟上的置換區域
- 字元串流的存取方式適合低速不定期的一連串字元 I/O 應用，如
  - 滑鼠
  - 數據機
  - 鍵盤(鍵盤的存取是典型字元串流介面的應用，使應用程式能取得或寫出一個字元，再加上緩衝與編輯的服務能從輸入串流中在插入一的字元，就可以輕易實作複雜一點的I/O)

45

區段與字元裝置

- 區段裝置介面包括所有存取磁碟機，及其它區段導向(block-oriented)裝置時所需的功能;最希望的是裝置能夠瞭解如read(), write()這樣的指令，如果它是一隨機存取裝置，會有一個seek()命令可以指定下一個將傳輸的區段
- 應用程式通常都透過檔案系統介面存取這類的裝置，像 read(), write(), seek()已抓住區段儲存裝置的主要核心特色，因此應用程式並不需知道這些裝置在處理低階動作時的不同。

46

網路裝置

- 網路插口介面(socket)在許多系統上被廣泛地使用，可想像成一個通用的電路插座插孔
- 網路插口介面使得應用程式能夠建立一個從插口連接到遠端主機的伺服程式，監聽任何遠端服務要求，並可經由插口傳送與接收封包資料

47

網路裝置

- 為支援伺服器實作，插口介面也提供select()函式來管理一組插口，一但呼叫該函式，便會回傳哪些插口有封包等待接收、有空間可傳送封包，可免去原本需要對網路I/O進行的輪詢或忙碌等待的工作
- 這些功能封裝了網路基本功能，對於建立使用各種網路硬體與協定的分散式應用程式有相當的助益

48

時脈與計時器

- 電腦的時脈與計時器提供 3 種基本功能：
  - 提供目前時間
  - 提供經過時間
  - 設定計時器在某時間觸發某動作
- 可程式化硬體計時器能夠設定一段時間後產生中斷，也可設定執行一次或重複執行這個程序以產生週期性的時間中斷

49

時脈與計時器

- OS的排程器是使用這種機制在時間切片結束時發生中斷，磁碟I/O子系統也使用此機制來週期性地更新快取緩衝區內的資料，並寫回磁碟
- OS也利用虛擬時脈提供計時器介面給使用者行程，以模擬許多軟體計時器。OS在核心或計時器驅動程式內維護一中斷串列，以觸發中斷的先後順序排列，並將此串列中第一個中斷時間設定給計時器，當計時器發生中斷時，核心會發出訊號給要求者，並設定下一次的計時器中斷

50

時脈與計時器

- 許多早期的電腦系統藉由時脈週期所產生的中斷頻率每秒18至60次，每周期需要16毫秒(1/60)到56毫秒(1/18)，這樣的解析度過於粗略
- 現今的電腦每秒可執行數百萬個以上的指令，但中斷觸發的精確度卻被計時器的解析度及維護虛擬時脈的額外負擔所限制。如果計時器還同時負責系統本身的時脈，則系統時間將變得非常不準確

51

時脈與計時器

- 大部分電腦的硬體時脈是由另一高頻計數器所負責，某些CPU(Pentium)有計數器供裝置暫存器讀取，這個與CPU同頻的高頻計數器，可以被視為高解析度的時脈，雖然這種時脈並不會產生中斷，但可以準確計量時間間隔
- 假設一部時脈速度為1000 MHz (megahertz) 的電腦且CPI (clock cycle per instruction)為5，則此部電腦的MIPS (million instructions per second)為多少？(A) 100 (B) 200 (C) 1000 (D) 5000

52

阻隔與非組隔 I/O

- 阻隔式 I/O
  - 系統呼叫時，此應用程式(AP)會停止執行，從作業系統的執行佇列被移到等待佇列；直到系統呼叫完成後，才將應用程式從等待佇列再移回到執行佇列繼續執行，並接收系統呼叫所回傳的資料
- 非阻隔式 I/O
  - 在系統呼叫後立即回傳 I/O 處理的狀態，即使 I/O 尚未完成也不等待

53

阻隔與非組隔 I/O

- I/O裝置實體的執行通常是非同步的，其花費時間變化很大而且不可預期，所以大部分OS對於裝置的應用程式介面都選擇使用阻隔式的I/O系統呼叫，因為比較容易了解
- 有些則必須使用非阻隔式I/O系統呼叫，例如：使用者介面一面接收鍵盤與滑鼠輸入，一面顯示資料於螢幕上；應用程式一面由磁碟檔案讀取影像資料，一面解壓縮顯示在螢幕上

54

阻隔與非組隔 I/O

- 有OS提供非阻隔式I/O系統呼叫，它不會長期地中斷應用程式直到系統呼叫執行完畢，只要目前沒有I/O進行，系統就馬上回應，讓呼叫者得知目前I/O進行的狀態，並不一定非要等I/O做完
- 多執行緒是另外一種達成 CPU 與 I/O 重疊執行的方法，使得在其他執行緒進行的同時可以執行阻隔式I/O系統呼叫。Solaris利用多執行緒，已經支援使用者模式的非同步I/O函式庫，不必再經由系統呼叫或是驅動程式，可減輕應用程式設計的負擔

55

阻隔與非組隔 I/O

- 另一種非阻隔式I/O系統呼叫的方法是使用非同步系統呼叫，它會在呼叫後馬上回應，行程不必等待I/O完成，而是繼續執行工作。在I/O完成工作後可以藉由數種不同的方式通知應用程式(設定一些共用變數、藉由訊號、軟體中斷、回授常式的觸發)
- 非阻隔式和非同步系統呼叫之間的差別是非阻隔式系統呼叫在讀取時，只要目前沒有I/O進行，就會立即回傳結果；而非同步系統呼叫在呼叫後馬上回應，且實際的I/O會同時繼續進行，直到完全結束後再通知呼叫者

56

核心 I/O 子系統

- 核心提供許多 I/O 的服務，這些服務都是建立在硬體與裝置驅動器的基礎之上，由核心的 I/O 子系統所提供
- I/O 排程的目的是要找出一個好的 I/O 執行順序來提昇系統的整體效率，公平分配數個行程所共享的裝置，並減少I/O 的平均等待時間

57

考題

- 下列何種網頁設計格式，主要用途是在Internet 上提供跨平台、跨程式的資料交換格式處理？
  (A) DHTML (B) HTML (C) VRML (D) XML
- 下列對於關聯式資料庫表格正規化 (normalization) 的敘述，何者錯誤？
  (A) 正規化的程度越高，資料的重複性會降低
  (B) 正規化的程度越高，資料存取效能亦會越高
  (C) 正規化的程度越高，資料表格的數量亦會增多
  (D) 正規化程序可避免更新異常

A1	*A2*
a3	b3
a4	b4

A1	*A2*
a1	b1
a2	b2
a3	b3

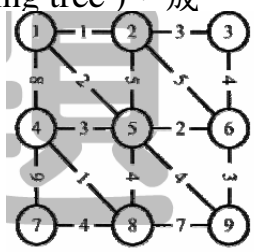
- 根據圖之結果，係屬於SQL 中的何種操作？
  (A) DIFFERENCE (B) INTERSECTION
  (C) JOIN (D) UNION

A1	*A2*
a1	b1
a2	b2
a3	b3
a4	b4

58

考題

- 圖之最小展開樹 (minimal cost spanning tree)，成本為何？ (A) 17 (B) 18 (C) 19 (D) 20



- 使用最大堆積(heap) 儲存數值資料，在此資料結構中搜尋最大數的時間複雜度為何？(A) O(1) (B) O(n) (C) O(log n) (D) O(n^2)
- 在演算法分析技術中，每次進行決策都是選擇目前最好的方向前進之解題方法稱為：

  (A) 貪近(greedy) (B) 各個擊破(divide - and – conquer) (C) 動態規劃(dynamic programming) (D) 修剪與搜尋(prune and search)

59

考題

- 檔案依存取方式的不同，可以分為那兩大類？

  (A) 輸入檔、輸出檔 (B) 主檔、交易檔
  (C) 循序檔、隨機檔 (D) 實體檔、邏輯檔
- 下列何種技術，允許CPU 在前一個指令尚未完成前，就開始處理下一個指令？ (A)管線技術(pipelining) (B) 複雜指令集技術(CISC) (C) 平行處理技術(parallel processing) (D) 精簡指令集技術(RISC)
- 美國環境保護署(Environmental Protection Agency, EPA)推出能源之星(Energy Star)的標準。下列電腦系統的功能，何者不是能源之星的標準？(A) 設定電腦休眠 (B) 設定關閉硬碟 (C) 設定螢幕保護 (D) 設定關閉螢幕

60

考題

- 雷射印表機是運用雷射光產生電荷，吸住碳粉，在紙上形成影像，其運作原理與下列何種設備相似？ (A) 數位相機 (B) 燒錄機 (C) 雷射光碟片 (D) 影印機
- RFID(radio frequency identification)是一種通過無線電波識別特定物品的技術，下列何者不是使用RFID技術的優點？(A) 可進行商品的追蹤 (B) 可增進商品銷售率 (C) 可降低商品失竊率 (D) 可使商品庫存盤點自動化
- 下列何者不是無線射頻識別(Radio Frequency – Identification; RFID)系統所具有的特色？ (A) 耐久 (B) 體積小 (C) 保護個人隱私 (D) 可主動提供資訊

61

考題

- 作業系統在記憶體有限的環境下，提供比實際記憶體還大的空間給程式使用。何者具備此種功能？ (A) 快取記憶體分段(cache memory segmentation) (B) 主記憶體鏈結(main memory linkage) (C) 輔助記憶體配置(secondary memory allocation) (D) 虛擬記憶體管理(virtual memory management)
- 下列何者可分擔部份CPU 的浮點計算工作，以提高系統的處理速度？(A) 磁碟控制器(disk controller) (B) 前端處理器(front - end processor) (C) 快速記憶體控制器(cache controller) (D) 協同處理器(coprocessor)

62

考題

- 假設三個程序 (process) 於時刻0秒同時到達，且其所需的CPU時間分別為5, 3, 4秒。若CPU在時刻12秒時可執行完此三個程序，在使用不同的CPU排程策略，則此三個程序的平均等待時間最少可為幾秒？ (A) 7/3 (B) 10/3 (C) 13/3 (D) 16/3
- 若一個程式的執行時間為100 log (n^2) + 2，則其最適合的時間複雜度表示方式為下列何者？ (A) O (log n) (B) O $(\log n)^2$ (C) O (n) (D) O (n^2)

63

考題

- 某個二元樹 (binary tree) 的前序式 (preorder) 為 ABDFGEC，中序式 (inorder) 為 FDGBAEC ，則其後序式 (postorder) 為何？ (A) FDGBECA (B) FGDBECA (C) FDGBCEA (D) FGDBCEA
- USB 2.0介面的傳輸速度最高是多少？ (A) 1.5 Mbps (B) 12 Mbps (C) 480 Mbps (D) 1100 Mbps
- 下列何者不是UML(unified modeling language)的一部份？(A) class diagram (B) collaboration diagram (C) entity-relationship diagram (D) sequence diagram

64

考題

- 藉由寬頻網路大量且迅速蔓延，致使網路癱瘓的病毒稱為：(A) 蠕蟲病毒(worm) (B) 巨集病毒(macro virus) (C) 特洛伊病毒(Trojan horse) (D) 千面人病毒(polymorphic virus)
- 下列何者並非資訊安全系統中，生物辨識裝置(biometric device)的技術？

  (A) 聲音辨識系統(voice recognition system)

  (B) 虹膜辨識系統(iris recognition system)

  (C) 個人身分認證系統(personal identification system)
  (D) 簽名辨識系統(signature recognition system)

65

I/O 排程

- 磁碟臂位於磁碟起始位置，有3個AP要對此磁碟進行阻隔式讀取，AP1要求讀取的區塊接近磁碟底部的位置， AP2要求讀取的區塊接近磁碟中間的位置， AP3要求讀取的區塊接近磁碟起始的位置，若OS將此I/O工作排定的服務順序為3➔2➔1，則可使磁碟臂移動距離最短，而減少AP等待的時間，所以排定I/O服務順序是I/O排程最重要的基本工作

66

I/O 排程

- 一般 I/O 排程的實作是讓每個裝置維護一個要求佇列，當應用程式要求一個阻隔式I/O 系統呼叫時，這個要求就會被放置在指定裝置的要求佇列中，再經由作業系統的I/O 排程器重新排定佇列內的執行順序

67

緩衝

- 緩衝區是一個介於兩個裝置或是裝置與應用程式間傳遞資料的記憶體區域
- 使用緩衝區的 3 個理由
  - 讓不同速度的生產者與消費者間傳遞資料
    - 數據機與硬碟速度不同，數據機用2個緩衝區交替寫入硬碟，雙重緩衝的機制不僅降低因生產者與消費者間的資料整理速度不同所帶來的問題，放寬時間限制
  - 調整裝置間不同大小的資料傳輸
    - 網路傳輸的資料常是不同大小，發送端將大量資料切割成小的網路封包傳遞，接收端則將收到的封包放置在重組的緩衝區內，組成原始資料

68

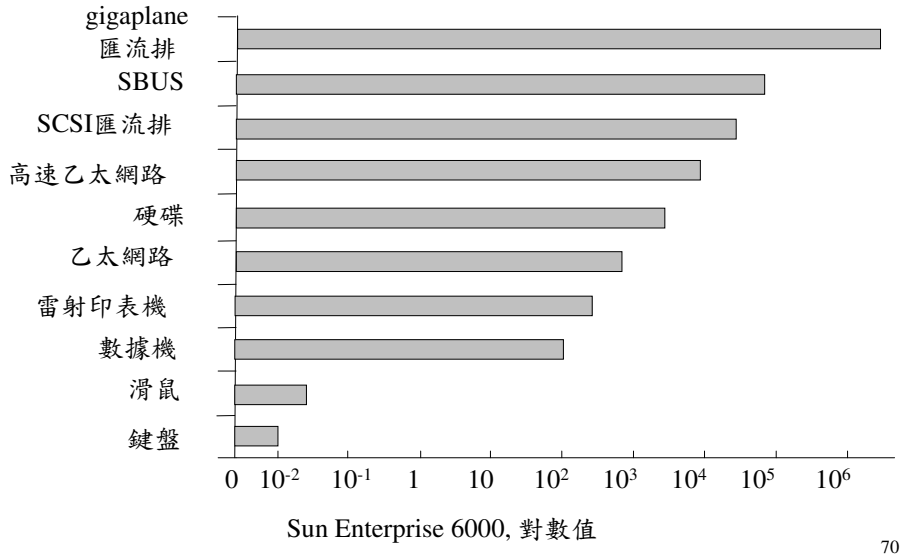
緩衝

– I/O 的拷貝語意

- 拷貝的結果不會因時間變化而有不同，AP想將緩衝區內的資料寫入磁碟時，會呼叫寫入系統呼叫，並提供指到此緩衝區的指標和寫入資料的大小。在系統呼叫回傳後，AP卻改變緩衝區內的資料內容，結果寫入的資料可能與原先呼叫時不同。若支援拷貝語意，則保證寫入資料與系統呼叫時相同，不會因後來的修改而不同。拷貝語意的實作是OS在接收到寫入的系統呼叫時，會將AP所要寫入的資料拷貝到核心緩衝區，再將控制權交還AP，因此寫入資料是從核心緩衝區來的，與AP對緩衝區修改的資料無關。但太多的複製會造成許多額外的負擔，所以減少複製次數是改進效率的重要關鍵

69

裝置傳輸速率



Sun Enterprise 6000, 對數值

70

快取

- 快取是先將資料複製到速度較快的記憶體中再行存取的方法，有些系統提供多層快取，其中最高與最低的速度差別很大 (WCPUID.exe)
- 目前正在執行的指令，原來存於硬碟，首次讀取後存於記憶體，並複製到CPU的快取，當在次使用相同指令時，可直接由快取讀取

71

快取

- 快取和緩衝是兩種不同的功能，緩衝只持有資料的拷貝，快取會將未在其他裝置的資料複製到速度較快的儲存裝置，不過有時兩者可能會使用到同一塊記憶體
- 為支援拷貝語意，使磁碟I/O更有效地排程，OS會利用主記憶體內的緩衝區來暫時保存資料，這些緩衝區也可以拿來當做快取。當核心接受一檔案I/O要求，會先於緩衝區內查看該檔案是否已經存在，如果存在，則不需要進行實體的磁碟I/O存取

72

週邊並行(Spool)和裝置預留

- 週邊並行，是指利用緩衝區暫存週邊裝置資料，使得不能夠多工處理的系統也能加強CPU與 I/O 的並行處理，以增進系統效能，印表機支援多人共用時，由OS伺服器精靈(daemon)或核心先攔截，在整理所有到印表機的輸出動作，應用程式的輸出會以磁碟檔案的型式先儲存於處理周邊並行的佇列中，當印表機印完一份資料後，周邊並行會一次從佇列中取出一個應用程式的檔案，再傳送到印表機，同時新的輸出會在存到該佇列

73

週邊並行(Spool)和裝置預留

- 週邊並行是作業系統提供與協調並行輸出的方法，尤其對磁帶和印表機等不能多路(multiplex)處理I/O要求的裝置
- 有些作業系統則提供裝置的互斥存取，讓行程以預留的方式配置及釋回裝置
- 有些作業系統強制只為每一裝置預留一開啟的檔案指標，或提供一些功能，使行程間能相互協調互斥存取

74

錯誤處理

- 使用保護模式的作業系統，能夠處理許多硬體與應用程式的錯誤，使得系統不會因為一點點的機械故障而造成整個系統無法運作
- 裝置與I/O傳輸在很多情況下都會發生錯誤，OS只能處理短暫性的錯誤，一般是再試一次發生錯誤的呼叫，若是永久性的錯誤，則只能更換硬體
- 一般錯誤處理的做法是檢查 I/O 系統回傳的狀態位元，也就是成功或失敗的訊息，當該位元狀態顯示失敗時，再進行錯誤處理程序(SCSI協定會提供非常詳細的錯誤狀況)

75

核心資料結構

- 系統可以透過核心資料結構，來了解 I/O 元件目前使用的狀況和其他的 I/O 活動
- UNIX提供各種檔案存取。讀取使用者檔案時，核心在執行磁碟I/O前要先檢測快取緩衝區是否已有資料，讀取基本裝置時，核心要確定所要求的資料大小是否為磁區的整數倍，以及是否對位整齊，讀取行程映像檔時，核心僅需從記憶體複製資料。

76

核心資料結構

- UNIX系統利用物件導向的方法，將這些檔案存取的相異處封裝在統一的結構中
- WINDOW NT對I/O 處理使用訊息傳遞的方法，將一個I/O要求轉換成一個訊息，由核心先傳遞給I/O管理者，再到裝置驅動程式，在傳遞的途中訊息內容都可以被更改，對輸出而言，該訊息包含寫入資料，對於輸入而言，該訊息則包含接收資料的緩衝區，這種訊息傳遞的方式與用共享資料結構的程式設計方法比較起來，能簡化I/O系統架構，並增加系統設計的彈性

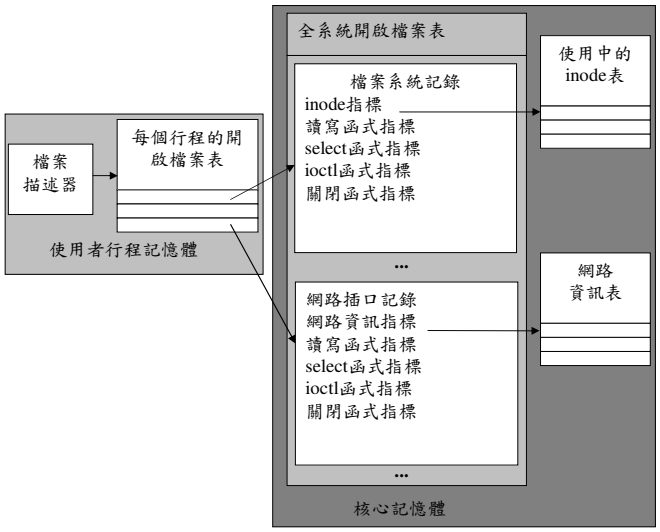
77

核心資料結構

- I/O 子系統負責協調可供應用程式與核心使用的服務，其主要負責的項目如下：
  - 檔案和裝置名稱空間的管理
  - 檔案和裝置的存取控制
  - 操作控制
  - 檔案系統空間配置
  - 裝置配置
  - 緩衝、快取、和週邊並行
  - I/O 排程
  - 裝置狀態監控、錯誤處理、和失敗回復
  - 裝置驅動器的設置與初始化

78

UNIX I/O核心結構



79

實作議題

- 設計 I/O 系統時需要考慮的議題
  - I/O 功能實作在硬體、裝置驅動器或是在應用程式之中
  - 裝置驅動器與應用程式間的操作轉換
  - I/O 的執行效率

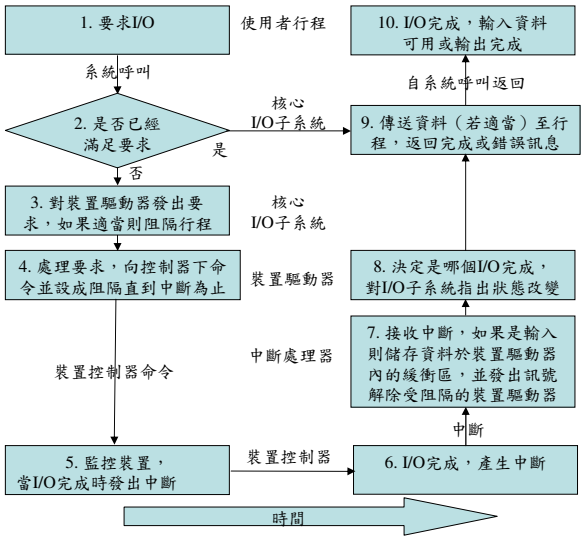
80

操作轉換

- 以磁碟讀取檔案來說，應用程式是透過檔案名稱讀取檔案內的資料，而檔案系統則必須先查詢檔案系統目錄中的檔案名稱，才能獲得檔案的位置，進而提供資料給應用程式

81

I/O 要求的生命週期



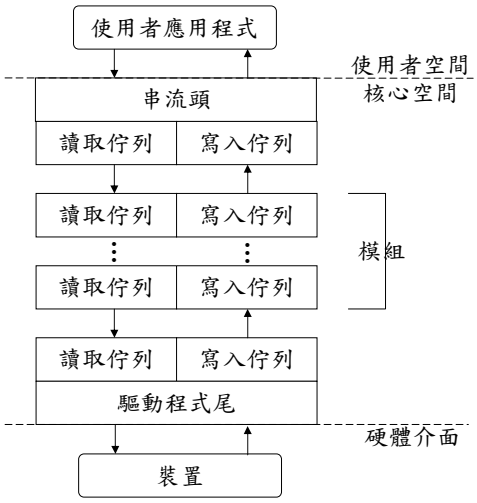
82

驅動器串流(Streaming)

- 串流是在裝置驅動器和使用者行程之間全雙工的連接管道
- 串流是由串流頭、控制裝置的驅動器尾、和中間零或多個成線性排列的串流模組所組成
- 驅動器串流 I/O 是一種非同步的 I/O

83

驅動器串流結構



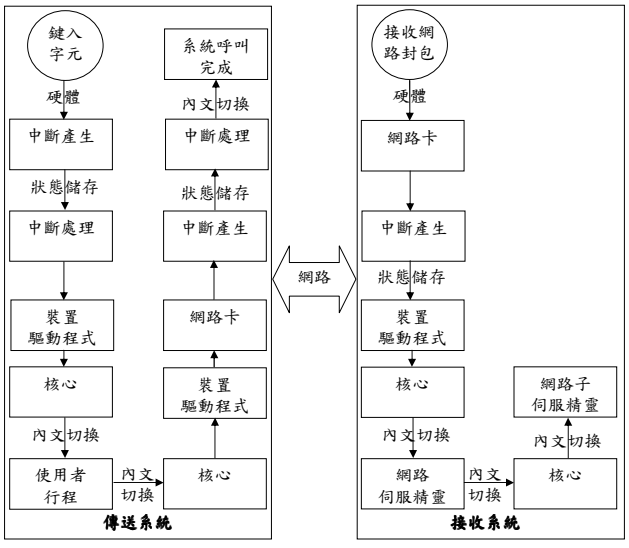
84

效率

- I/O是影響系統效率的主因
- 改進 I/O 的執行效率
  - 減少內文切換的次數
  - 減少資料複製到記憶體上的次數
  - 藉由大量傳輸、智慧型控制器與輪詢減少中斷的頻率
  - 增加使用 DMA 控制器的平行處理
  - 減少 CPU 對於簡單資料複製處理的次數
  - 將初始條件移到硬體上處理平衡 CPU、記憶體子系統、匯流排與 I/O 的執行效率

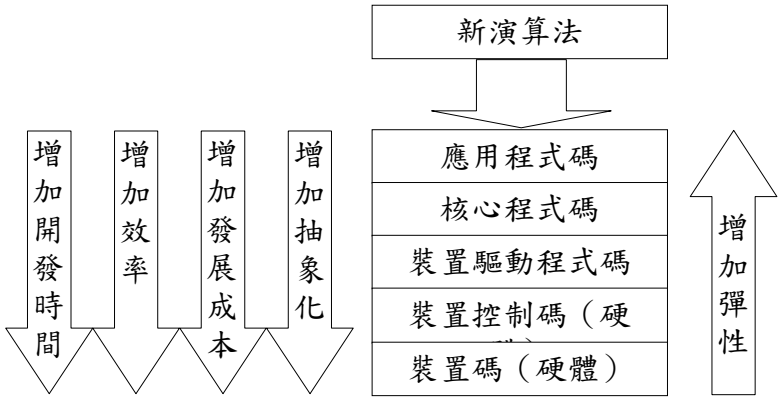
85

電腦通訊



86

裝置功能進行階程



87

摘要

- I/O 的基本元件
  - 匯流排
  - 裝置控制器
  - 裝置本身
- 系統呼叫介面提供裝置服務給應用程式使用，包括
  - 區塊裝置
  - 字元裝置
  - 記憶體對映檔
  - 網路插口
  - 計時器

88

摘要

- 核心 I/O 子系統提供數種服務
  - I/O 排程
  - 緩衝
  - 週邊並行
  - 錯誤處理
  - 裝置保留
- 實作議題
  - 操作轉換
  - 驅動器串流
  - 效率

89