類別: 計算機概論

題目: Multi-tasking與相關的幾個概念

副標題: 〈同時點開好多個程式，究竟電腦是怎麼處理的呢？〉第二回

內容

上一回提到作業系統中的「排程功能」。這一回將要來介紹「Multi-tasking」的概念，以及電腦執行多個程式會順順的或是卡卡的原因。

Multi-tasking

Multi-tasking的概念常常跟其他幾個名詞搞混，分別有：

1. Multi-programming
2. Multi-processing
3. Multi-tasking
4. Multi-threading

好像有點暈頭了(⊙\_⊙)？沒關係，讓我們從每個名詞它想達成的目的來了解應該會比較清楚。

Multi-programming: 給我每分每秒的工作，不准休息！

假設電腦正在跑一個程序(程序A)，此時這個程式需要使用者輸入一段文字，然後才能進行接下來的工作。萬一使用者遲遲不寫點東西然後按Enter，那麼，一台不支援multi-programming的電腦就只能乖乖的等，所有的硬體資源例如CPU、記憶體等等通通都閒置下來，而其他等著被執行的程序通通都得等它一個，可想而知這是多麼的浪費資源。所以當一台有支援multi-programming的電腦遇到上面的狀況時，它就會把這個程序先擺到一邊，然後把硬體資源交給下一個等著被執行的程序(程序B)使用。記憶體裡面開始載入程序B所需要的資料，然後CPU開始執行程序B的運算。這個轉換的過程叫做「context switching」。透過multi-programming的運作，系統資源就能夠被好好的利用，減少它們處於閒置的狀態。

複習一下，這個程序A在等著I/O的時候會變成Waiting state，等到I/O完成之後再度回到Ready state，等著被CPU scheduler再次選中。」

Multi-processing: 所有的CPU都動起來，趕快把該做的工作給做完！

要有multi-processing的前提是電腦需要有2個以上的CPU。一個程序有能夠大家同時執行的部分，也有需要等別人處理完之後自己才能用的部分，就好比有多個人擁有一個共同的銀行帳戶，當執行「存款」動作的時候，大家同時存錢並不會造成困擾，但是當大家同時「提款拿來花」的時候，可能就會造成某某A把所有錢拿去買賓士，而某某B把所有錢拿去買重機，結果發現存款早就被A花光，付不出錢的窘境。

針對能夠大家同時執行的部分，multi-processing就能夠發揮它的效用，多個CPU各自認領幾個「存款」的動作，然後大家一起同時執行，這樣就能夠早一點把所有「存款」的動作給做完，讓一個程序能夠早一點被完成。

有multi-processing的系統有幾個優點：

1. 能夠更快的完成工作(當然～這就是主要的目的)
2. 具有經濟效應：記憶體、周邊元件等等可以看做是程序執行的成本，那麼有越多CPU同時執行的話，那平均每個CPU運作所需的成本就會降低。
3. 增加可靠性：有多個CPU可以用的話，那即使其中幾個CPU壞掉，程序還是能正常執行，只是比較慢而已。

Multi-tasking: 快速的切換，讓人以為多個程序是「同時」被執行

這個就是讓電腦能夠一邊播放音樂一邊讓妳用word或是ppt的秘密了！在電腦支援multi-programming的前提下，給每個程序的執行分配一小段時間，時間一到就把當前的程序暫停，然後做context switching，完了之後開始執行另一個程序。同樣的時間一到就再換下一個程序執行。這個「時間一到就換下一個人」的方式叫做Time sharing。所以multi-tasking是透過time sharing加上context switching來實作的。

Multi-threading: 欸欸欸，給點回應吧？

首先要來了解一下什麼是thread。Thread又叫做執行緒，就是一組要被執行的步驟。一個程序至少由一個執行緒(主執行緒)構成，並且能夠視情況再增加thread去執行其他任務，來保持主執行緒能夠正常運行，維持程序的回應性。舉個簡單的例子，假設有一個網頁伺服器沒有採用multi-threading，當伺服器收到用戶A傳來想要看某某網站的request時，此時它(主執行緒)就開始處理，而萬一這時用戶B也傳來想要看某某網站的request，但因為伺服器正在忙著處理用戶A的事情，所以就沒辦法回應用戶B，讓用戶B收到「伺服器忙線中」之類的結果，得等到伺服器處理完用戶A的事情之後才能回過頭來處理用戶B的request。因此在用戶B看起來，網頁伺服器好像是「沒有回應」。

Multi-threading的好處在於當用戶A傳來一個request後，伺服器就開起一個新的thread，讓它專門處理用戶A的事情，而伺服器本身就繼續監聽有沒有新的用戶傳來request。當用戶B傳來request時，因為用戶A的事情已經被分配下去執行，所以伺服器能夠接收用戶B的請求，同樣的開一個新的thread去處理用戶B的事情，並繼續監聽有沒有新的用戶發出request。這樣子伺服器就能夠一直處理新用戶的請求，來保持它的回應性。

電腦卡卡的原因？

在上一回文章有提到，電腦同時開著Line、youtube、word或ppt，可能會有「跑起來順順」或「跑起來卡卡」的情況，這很有可能跟記憶體的大小有很大的關係。能夠「同時」執行三個程式，靠的是multi-tasking，也就是Time sharing加上Context switching，白話一點就是說「時間一到就把系統資源交給下一個程序使用」。當電腦的記憶體太小的時候，裡面能夠存的東西大概只能夠放一個程序需要的資料，所以在做context switching時就得花更多的時間去做資料的置換，然後才能真正的開始執行下一個程序，感覺起來就會卡卡的。假如記憶體夠大，裡面已經存了每個程序需要的資料，那麼在不同的程序間切換時就可以省略去儲存空間撈資料的時間，讓多個程式能夠流暢的運作。

小結

跟作業系統有關的4個常常被搞混的概念其實從目的性來看應該就能夠很容易區分：multi-programming是為了增加系統資源的利用率、multi-processing是為了能夠更快的把該跑的程序執行完畢、multi-tasking是為了讓多個程序都能夠被「類同時」執行、multi-threading是為了維持程序的回應性。作業系統正是靠著實現上面的幾個概念來讓電腦能夠快速有效的運作！

參考資料

Difference between multiprogramming, multitasking, multithreading and multiprocessing

<https://www.geeksforgeeks.org/difference-between-multitasking-multithreading-and-multiprocessing/>

如果卡卡的原因!

8/12,8/27

8/12 (五), 無痛, 前一天得吃流質食物, 前2天吃低渣食物(

去一葛地方? 文房