Chap4 各種 CPU 排程演算法之探討

一、先進先出排程法(First in First out；FIFO)

按進入 Ready Queue 的順序而決定使用 CPU 的先後次序而完成此方式是採用 FIFO Queue。

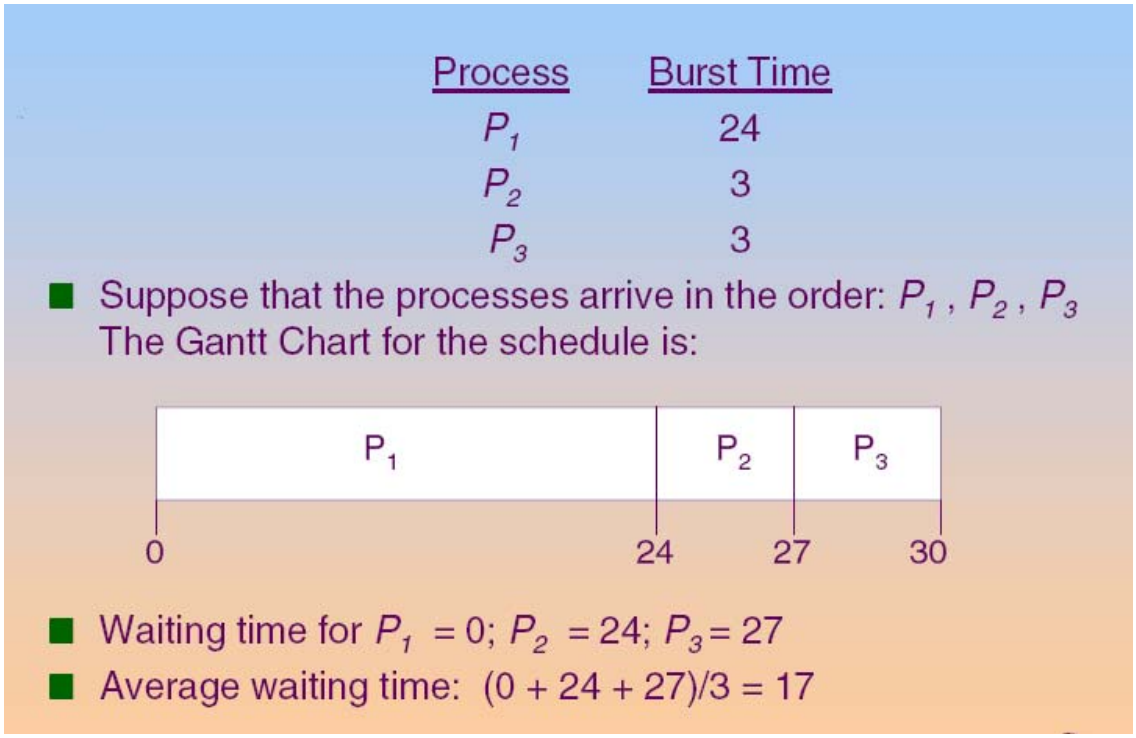
特點有三:

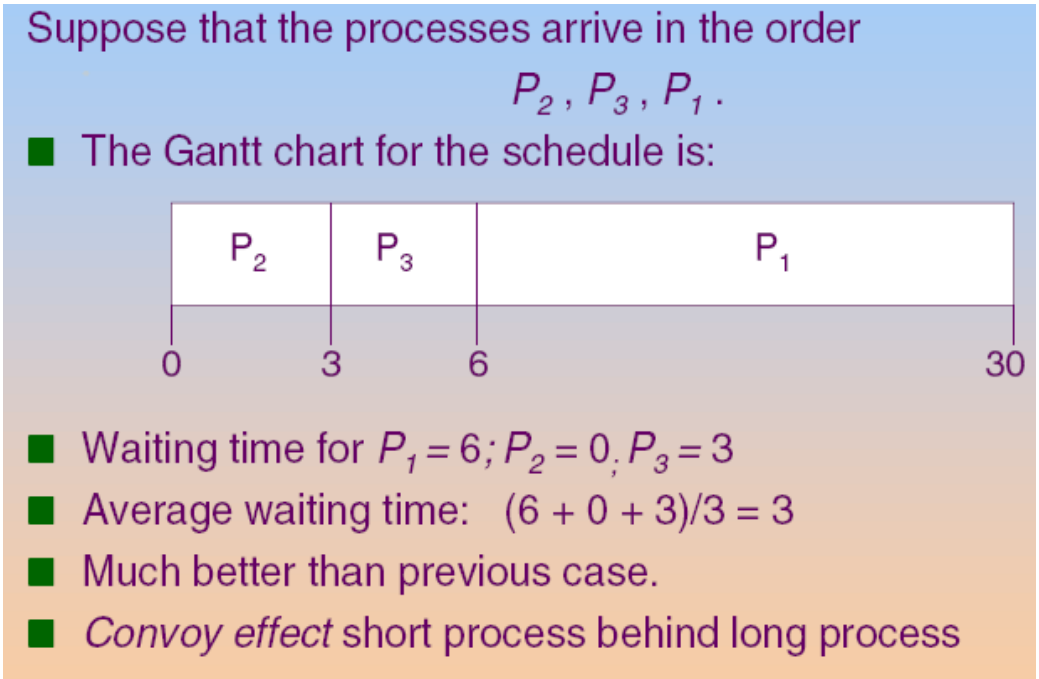
(1) 此種排班程式最容易設計。

(2) 此種排班程式的效益最差，即平均等待時間較差。

(3) 可能會造成護送效應 (Convoy Effect): 很多短時間的 process，都在等一個長時間的 process 時，所產生的效應。(因爲等待時間很長)。

FIFO 排班程式是不可插隊 (Non-preemptive)，一旦一個 process 佔用了 CPU，它就會一直使用直到終止或請求 I/O 動作，如果允許某 process 長時間佔用 CPU，使用者會無法忍受，故不適用於 Time-sharing system。





二、最短程式優先排班法(Shortest Job First Scheduling; SJF)

此排班法會在 CPU 空閒時選擇具最小的 CPU Burst Time 的 process，並使之取得 CPU 控制權。

特點:

(1) SJF 是效益最佳的排班法。

(2) 由於各 process 的下一個 CPU burst time 較難預測，因此 SJF 很難被用在 Process Scheduler。

(註) SJF 又分爲可插隊(Preemptive)及不可插隊(Non- Preemptive)

(1) Preemptive Scheduling

意義: 當 process 獲得 CPU 後，雖然該 process 尚未執行完畢時，卻允許被移走(離開 CPU)。

適用: Real-time 及 Interactive system。

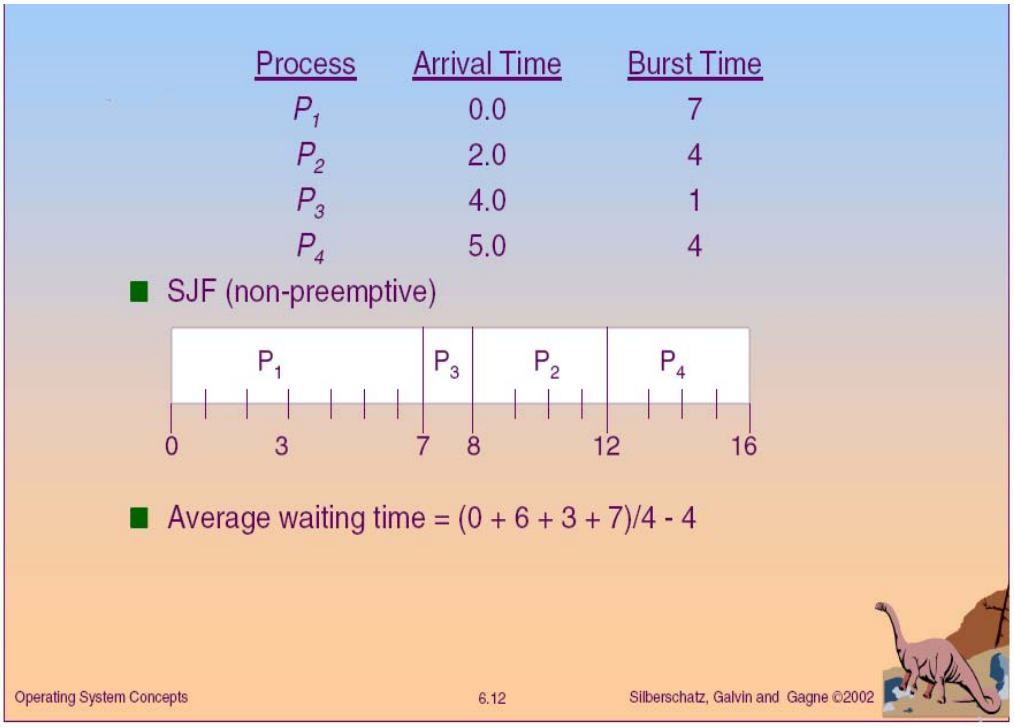
缺點: Context Switch 可能頻繁，浪費時間；可能會造成 Starvation。

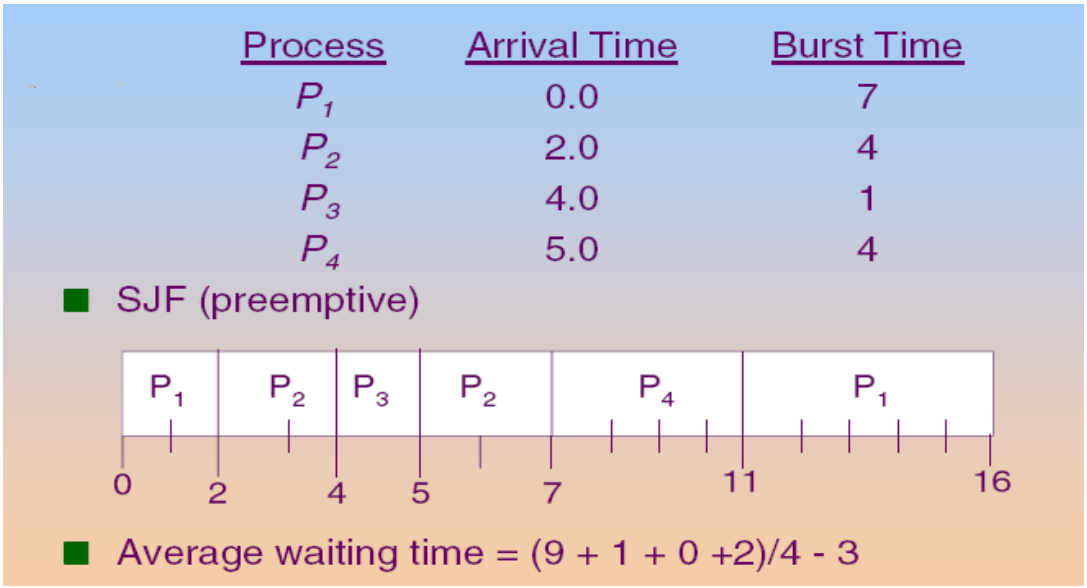
(2) Non-Preemptive Scheduling

意義: 當一 process 獲得 CPU 後，除非該 process 已經執行完畢，否則不允許被移走。

優點: 對所有的 process 較公平。

缺點: 平均等待時間可能會提高，易造成 Convoy Effect。





三、優先等級(Priority)排班法

優先等級排班法會將 CPU 指定給具有最高優先等級的 process。若同時間有多個相等優先等級的 process 欲使用 CPU 時，則採用 FIFO 方式。

Priority Scheduling 的一些問題: 飢餓現象(Starvation)：

優先等級排班法有可能使某些較低優先權的 process 限於無限期等待 CPU 的地步。一些高優先權 process 可能讓某些低優先權的 process 永遠得不到 CPU。

解決方式: Aging Technique。

對於低優先權 process 無限停滯問題的解決方法是老化(Aging)------而此技術可將

在系統中等待了很長時間的 process 的優先權逐漸提高。

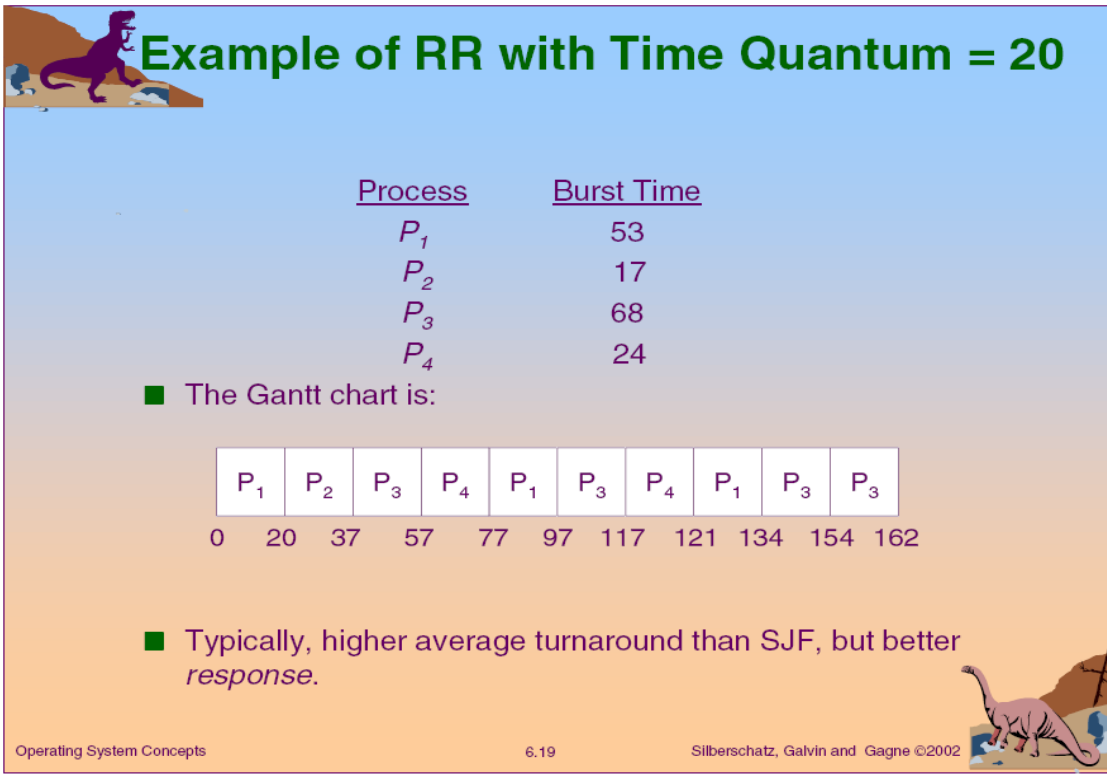
四、巡迴式排程法(Round-Robin Scheduling; R.R.)

R.R.排程演算法是特別爲 Time-sharing 設計的。在這種設計中，把一小段時間定義爲時間配額(Time Quantum)或時間分槽(Time Slice)，每個 process 皆有固定時間量來使用 CPU，當 process 使用 CPU 超過一個 Time Slice 時，則計數器(Timer)會產生中斷促使此 process 從 Running state 回到 Ready state。

<註> R.R. Scheduling 是 preemptive 的。R.R.演算法的效率與 Time Slice 的大小有很大的關係。

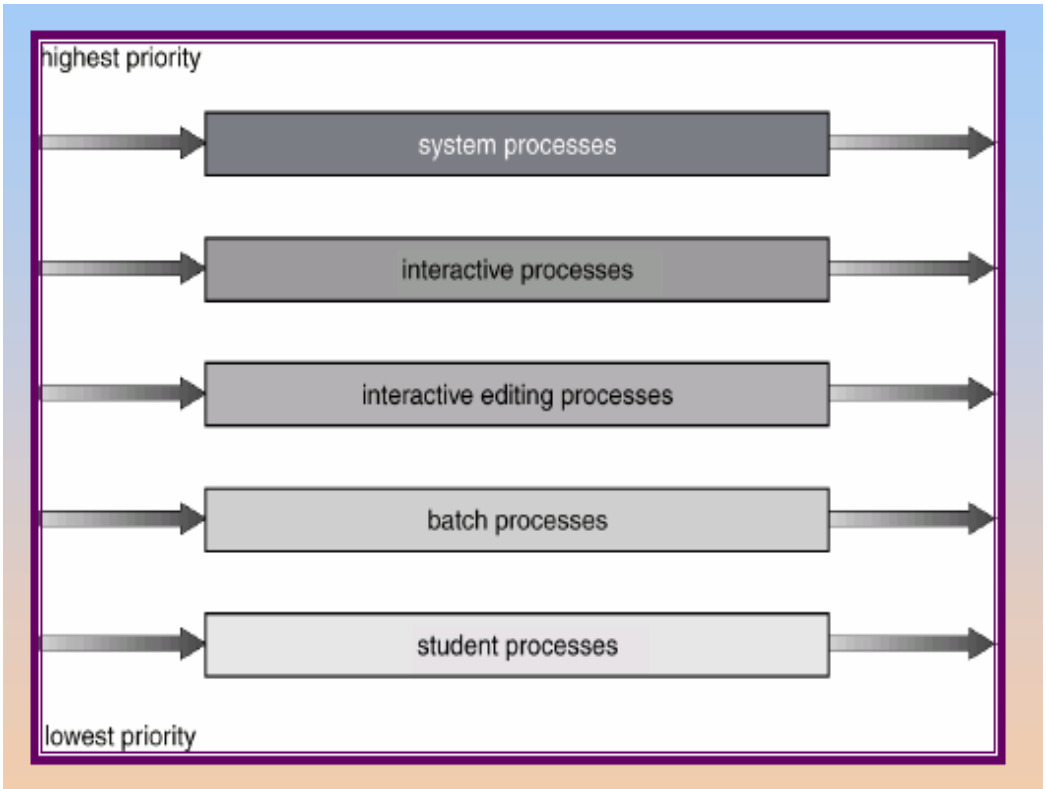
若 Time Slice 無限大，則 R.R.策略約等於 FIFO。

若 Time Slice 無限小，則 R.R.策略約等於 Time- Sharing System。



五、多重佇列排程法(Multiple-Level Queues Scheduling)

多重佇列排程法，是將 ready queue 進一步分成幾個不同 queue，根據 process 的某些特性，如 process 的型態，固定地將 process 分配到 queue 之中，每個 queue 又有各自的排班法則；例如，佇列可以用前景(foreground)與背景(background) process，對於 foreground process queue 使用排程，對於 background process queue 則使用 FIFO。此外，在各佇列之間，還必須進行排程，通常是一固定優先權可插隊的排班法則。



六、多重佇列回饋排程法(Multiple-Level Feedback Queue Scheduling)

通常，在多重佇列排程法中，process 在它進入系統時就會被永久地分配到一個 queue，process 不能在各個 queue 之間移動。

多重佇列回饋排程法，則可以允許 process 在 queue 之間移動，當某一 process 若在低優先權 queue 中等待了很長時間，系統可以將它移到一個高優先權的 queue 上，而這是一種老化(Aging)技術的形式，可以防止肌餓(Starvation)現象的發生。