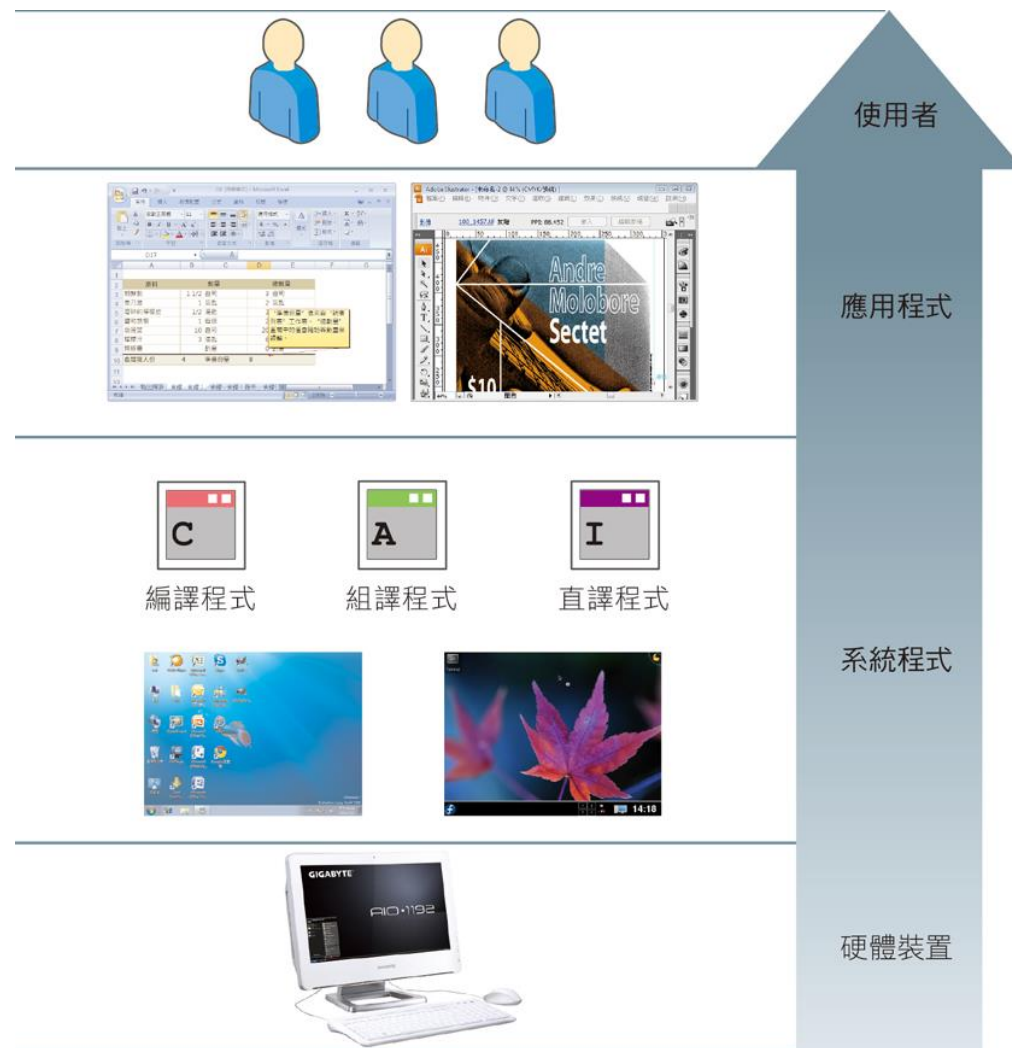


# 系統軟體

# 電腦系統的整體架構

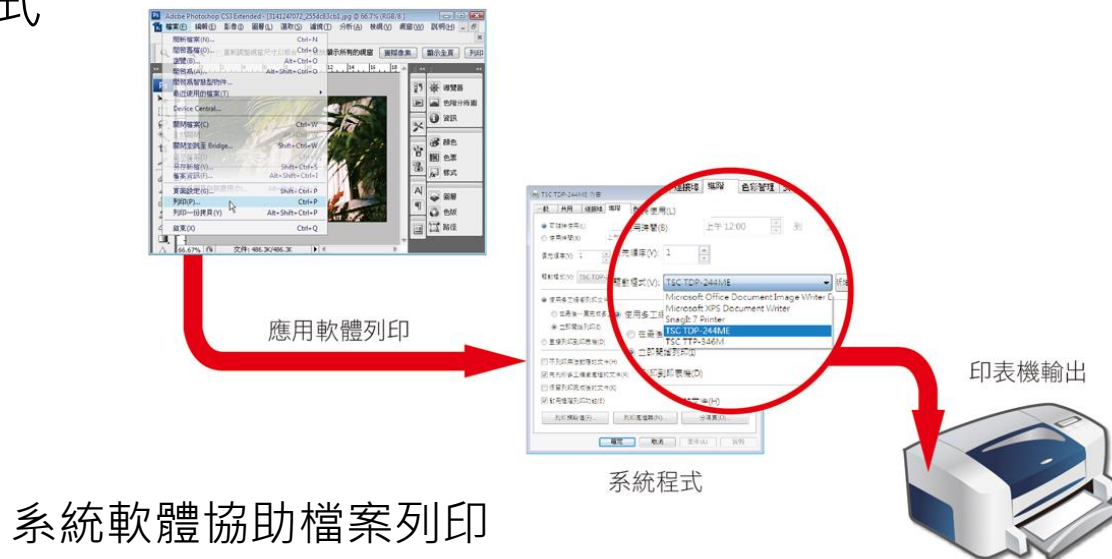
- ◆ 由外而內依序可分為
  - ◆ 使用者 → 應用程式 → 系統程式 → 硬體
- ◆ 應用程式必須靠系統程式的協助，才能快速有效的使用硬體資源



電腦系統的整體架構

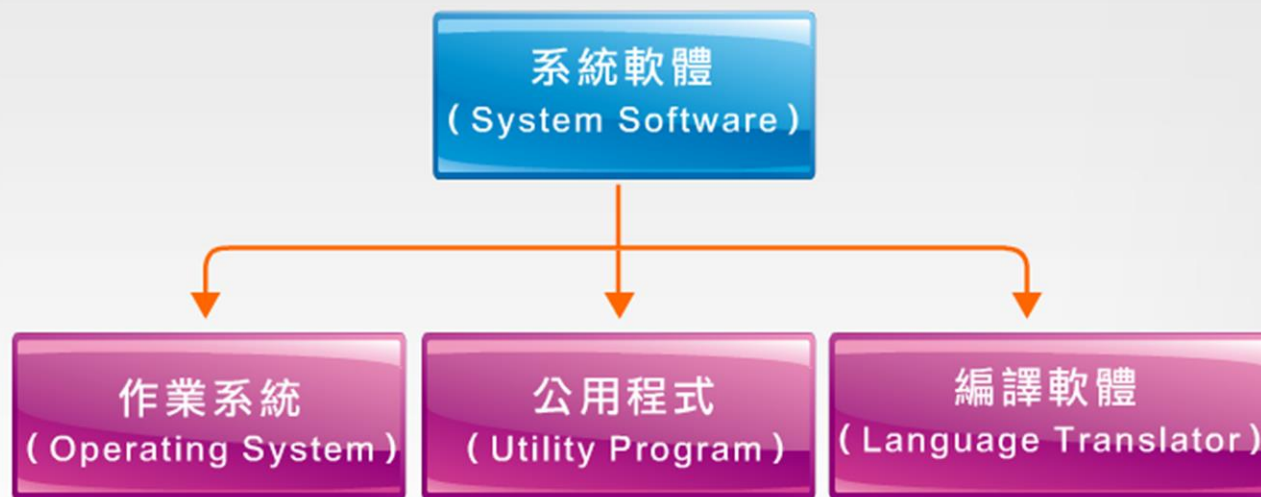
# 系統軟體 System Software

- ◆ 協調並管理硬體內的各項資源，如CPU、記憶體等
- ◆ 由各種支援電腦運作的程式所組成，讓使用者專注於應用程式的使用，而不必了解電腦硬體的運作原理
- ◆ 如：列印檔案時，作業系統和驅動程式等系統軟體協助應用軟體，將檔案轉換成印表機所能處理的資料格式



# 各種類型的系統軟體

- ◆ 系統軟體包含作業系統、網路軟體、發展工具和程式語言軟體、資料庫管理軟體等。
- ◆ 發展工具和程式語言軟體主要提供使用者程式設計的支援，包含以下各種：
  - ◆ 組譯程式 (Assembler)
  - ◆ 編譯程式 (Compiler)
  - ◆ 直譯程式 (Interpreter)



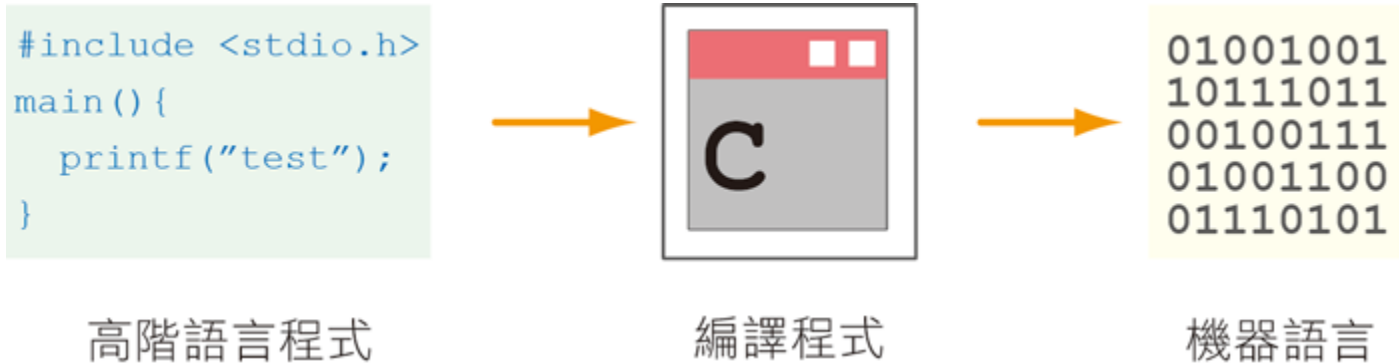
# 組譯程式 Assembler

- ◆ 電腦硬體只理解機器語言，但編寫及閱讀機器語言非常困難，所以組合語言採用助憶符號，便於程式撰寫
- ◆ 組合語言必須先用組譯程式轉譯成機器語言，才能執行



# 編譯程式和直譯程式

- ◆ 組合語言的撰寫與維護仍不容易，於是發展高階程式語言
- ◆ 高階程式語言須透過編譯程式或直譯程式轉譯成機器語言才能執行



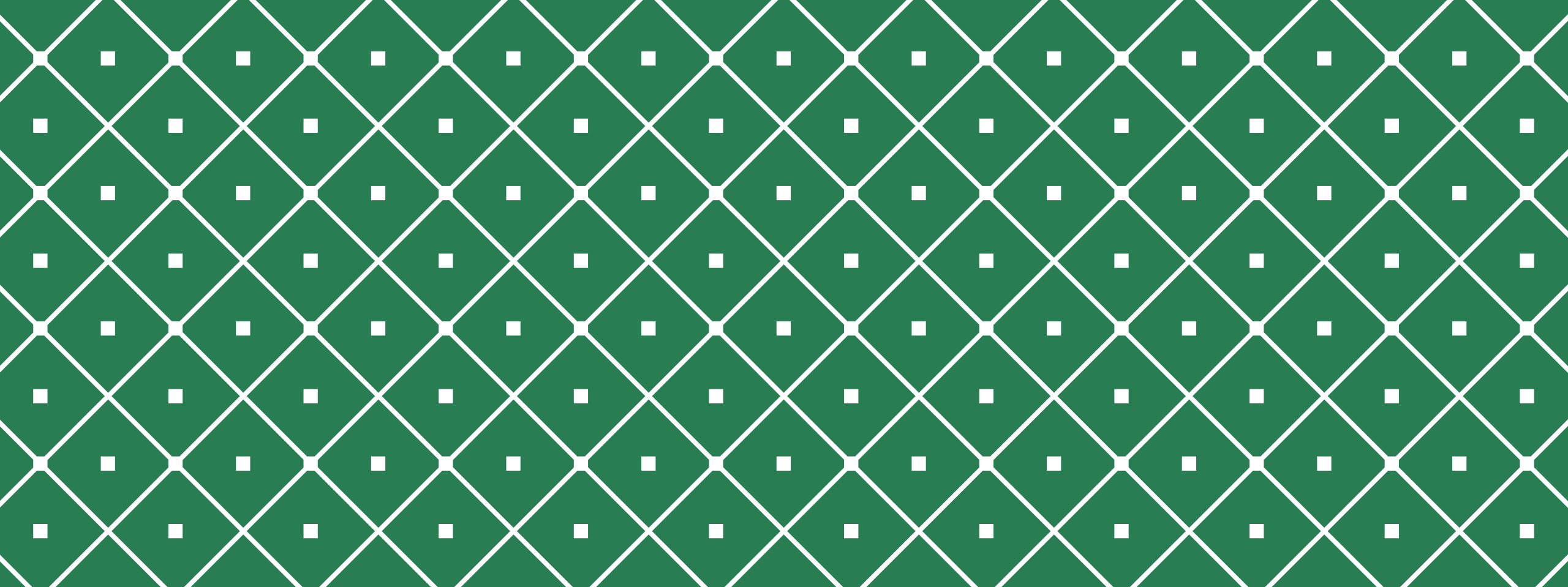
# Compiler VS Interpreter

## ◆ 編譯程式 Compiler

- ◆ 將高階程式語言所寫的程式，**整個**轉譯成機器語言後再執行
  - ◆ 如 C, C++, Java 等
- ◆ 適用於產生**可直接執行的檔案**
  - ◆ 如電腦上許多副檔名為 .exe 的執行檔，都是經過編譯後的程式

## ◆ 直譯程式 Interpreter

- ◆ **逐行執行**，每轉譯一行程式，就立刻執行，然後再轉譯下一行，再執行，直到程式結束或出錯為止
  - ◆ 如 JavaScript, PHP, ASP 等
- ◆ 適用於讓使用者**立即看到效果**的應用
  - ◆ 如使用 JavaScript 設計網站的文字效果、跑馬燈等



# 作業系統



# 開機程序

1

## 供應電力

電源供應器將電源傳送給主機板上的元件。

2

## 自我測試

執行 BIOS 的程式，讓硬體進行自我測試。

3

## 比對測試結果

CPU 比對硬體測試的結果是否與 BIOS 的資料相同。

5

## 載入作業系統

CPU 執行開機程式，將作業系統載入記憶體中。

6

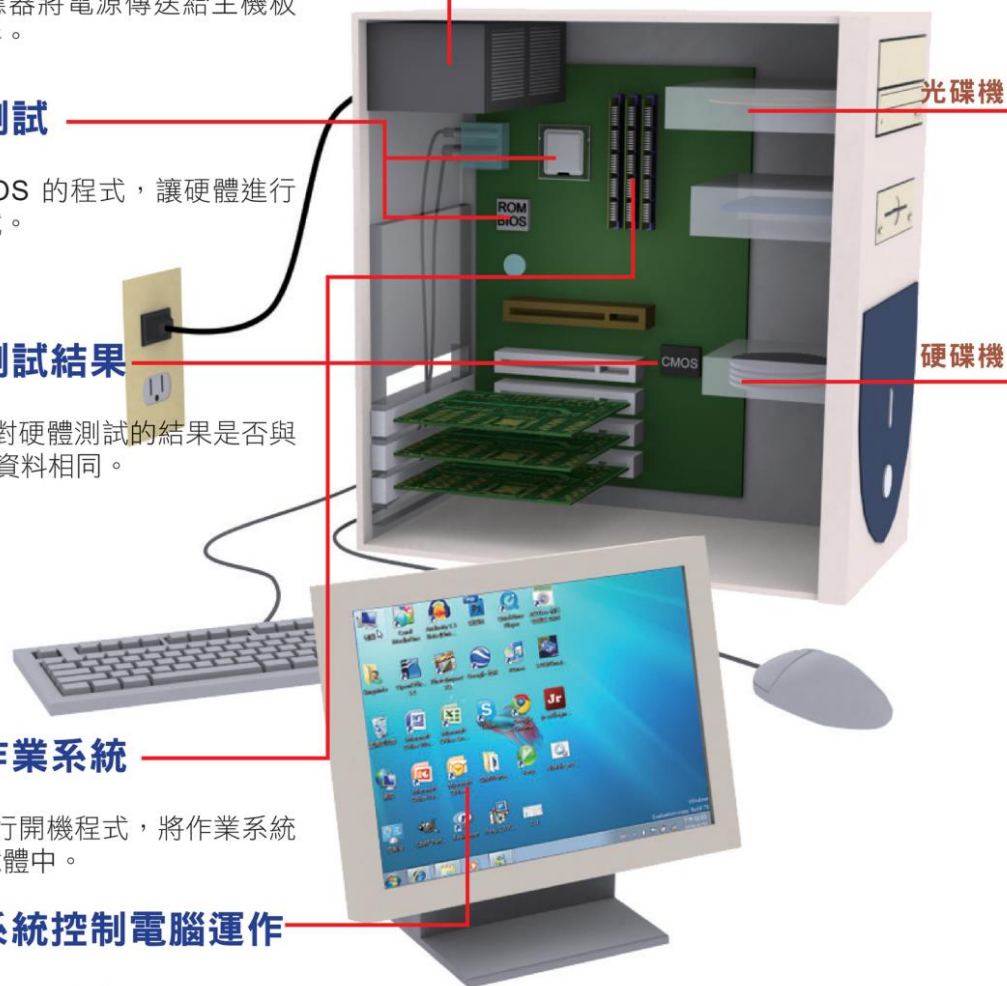
## 作業系統控制電腦運作

作業系統開始控制電腦運作，並將畫面顯示在螢幕上。

4

## 尋找開機擋

CPU 繼續執行 BIOS 程式，並於硬碟、或光碟中尋找作業系統的開機程式。



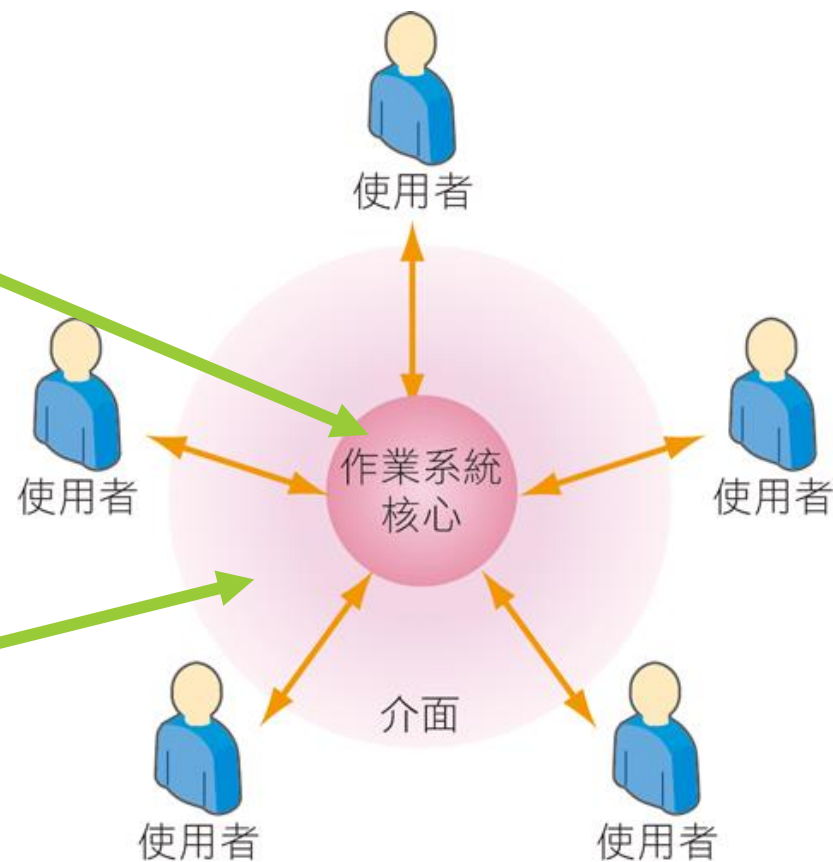
# 作業系統的重要性

- ◆ 指揮及協調對於硬體與軟體或應用程式等資源的管理
  - ◆ 處理軟體和硬體間的溝通
  - ◆ 處理軟體和軟體間的溝通
  - ◆ 同時執行多個軟體
- ◆ 讓CPU判斷邏輯與運算數值
- ◆ 讓主記憶體載入/讀出資料與程式碼
- ◆ 硬碟存取、網路卡傳輸資料、週邊設備運轉
- ◆ 何謂作業系統？

# 介面 Shell

- **核心**是作業系統最重要的部分
- 包含許多**軟體元件**，具有處理、分配及管理資源的功能

- **介面**將使用者的**命令**轉譯為核心認識的**程式碼**



# 文字模式介面 vs 圖形使用者介面

- ◆ 早期作業系統的介面是文字模式，使用者透過鍵盤溝通
- ◆ 目前有些作業系統仍採以文字模式介面為主，如 Unix 等
- ◆ 第一個圖形使用者介面的作業系統在 1980 年代出現
- ◆ 目前最常見的就是 Windows 以及 macOS，其程式、資料夾、檔案等，都使用小圖示 (Icon) 表示

```
drwxrwxr-x 2 root operator 512 Apr 16 07:30 .snap
-r--r--r-- 1 root wheel 6188 Feb 24 2008 COPYRIGHT
drwxr-xr-x 2 root wheel 1024 Apr 16 07:30 bin
drwxr-xr-x 7 root wheel 512 Apr 16 07:33 boot
drwxr-xr-x 2 root wheel 512 Apr 16 07:30 cdrom
lrwxr-xr-x 1 root wheel 10 Apr 16 07:33 compat -> usr/compat
dr-xr-xr-x 4 root wheel 512 Apr 20 12:56 dev
drwxr-xr-x 2 root wheel 512 Apr 16 07:30 dist
drwxr-xr-x 3 root wheel 512 Apr 19 20:02 downloads
drwxr-xr-x 24 root wheel 2560 Jul 15 19:12 etc
lrwxr-xr-x 1 root wheel 8 Apr 16 07:35 home -> usr/home
drwxr-xr-x 3 root wheel 1536 Feb 24 2008 lib
drwxr-xr-x 2 root wheel 512 Apr 16 07:30 libexec
drwxr-xr-x 2 root wheel 512 Feb 24 2008 media
drwxr-xr-x 2 root wheel 512 Feb 24 2008 mnt
dr-xr-xr-x 2 root wheel 512 Feb 24 2008 proc
drwxr-xr-x 2 root wheel 2560 Apr 16 07:30 rescue
drwxr-xr-x 4 root wheel 512 Jul 15 19:24 root
drwxr-xr-x 2 root wheel 2560 Apr 16 07:30/sbin
drwxr-xr-x 2 root wheel 512 Apr 19 23:09 service
lrwxrwxrwx 1 root wheel 11 Apr 16 07:30 sys -> usr/src/sys
drwxrwxrwt 10 root wheel 512 Jul 15 19:06 tmp
drwxr-xr-x 18 root wheel 512 Apr 16 20:47 usr
drwxr-xr-x 32 root wheel 512 Apr 29 01:41 var
```

# 常見的作業系統

硬體	研發公司	作業系統名稱
個人電腦	Microsoft	Windows 10, Windows 8, Windows 7, Windows XP
	Apple	Mac OS X
	自由軟體	Linux
伺服器	Microsoft	Server 2013, Server 2008, Server 2003
	Apple	Mac OS X Server
	自由軟體	Unix, Linux, FreeBSD
行動裝置	Microsoft	Windows Phone, Windows Embedded
	Apple	iOS
	其它	Google Android, Linux, Palm OS, Symbian

# 核心 Kernel

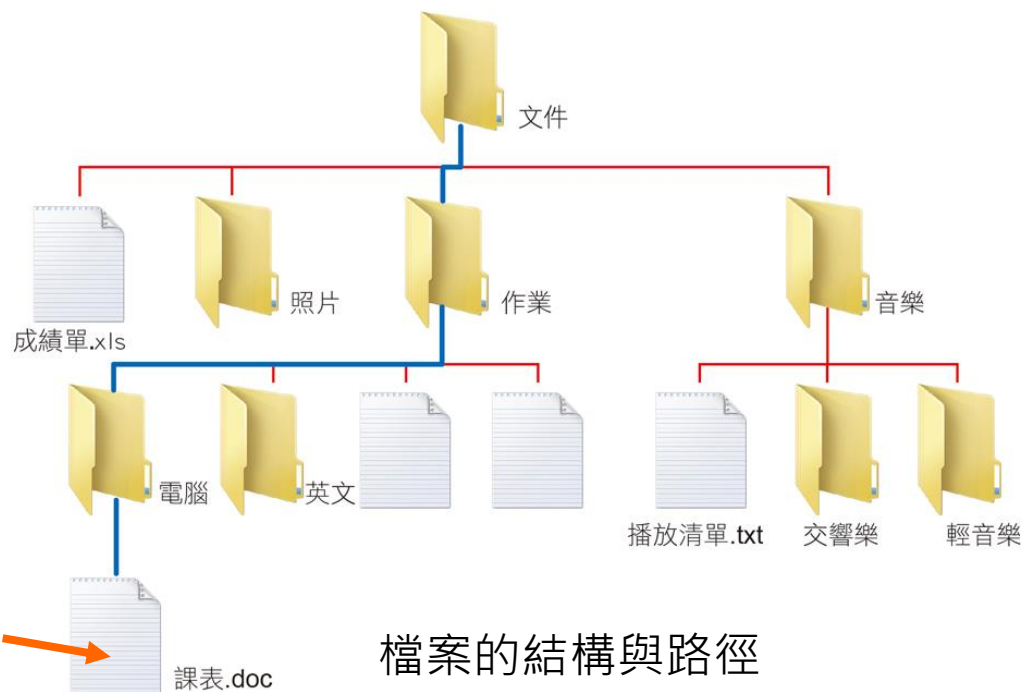
- ◆ 作業系統的核心功能包含
  - ◆ 檔案管理
  - ◆ I/O管理
  - ◆ 記憶體管理
  - ◆ CPU 行程管理

# 檔案管理

- ◆ 檔案是指資料儲存的集合，電腦檔案通常包含程式檔和資料檔
- ◆ 主要是管理輔助記憶體<sub>輔助記憶體</sub>的檔案和資料夾
- ◆ 許多作業系統的檔案結構都是以樹狀（ Tree ）結構呈現，如 Windows, Unix, Linux 等
- ◆ 檔案管理包含檔案的表示<sub>表示</sub>、檔案的操作<sub>操作</sub>、檔案的保護<sub>保護</sub>等

# 檔案的表示

- ◆ 資料夾或目錄是指以群組方式呈現的檔案集合
- ◆ 路徑是指由特定目錄到達特定檔案，所經過的資料夾及檔案名稱



文件\作業\電腦\課表.doc

檔案的結構與路徑

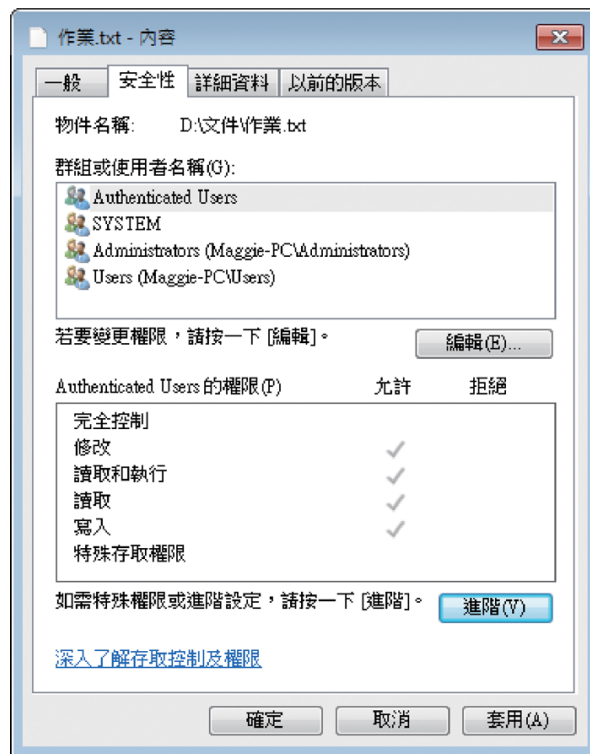


# 檔案的操作

- ◆ 檔案**操作**包含建立與刪除檔案、將資料寫入檔案、從檔案讀取資料、更改檔案的屬性等
- ◆ 檔案的**屬性**包含檔案的大小、日期、型態等

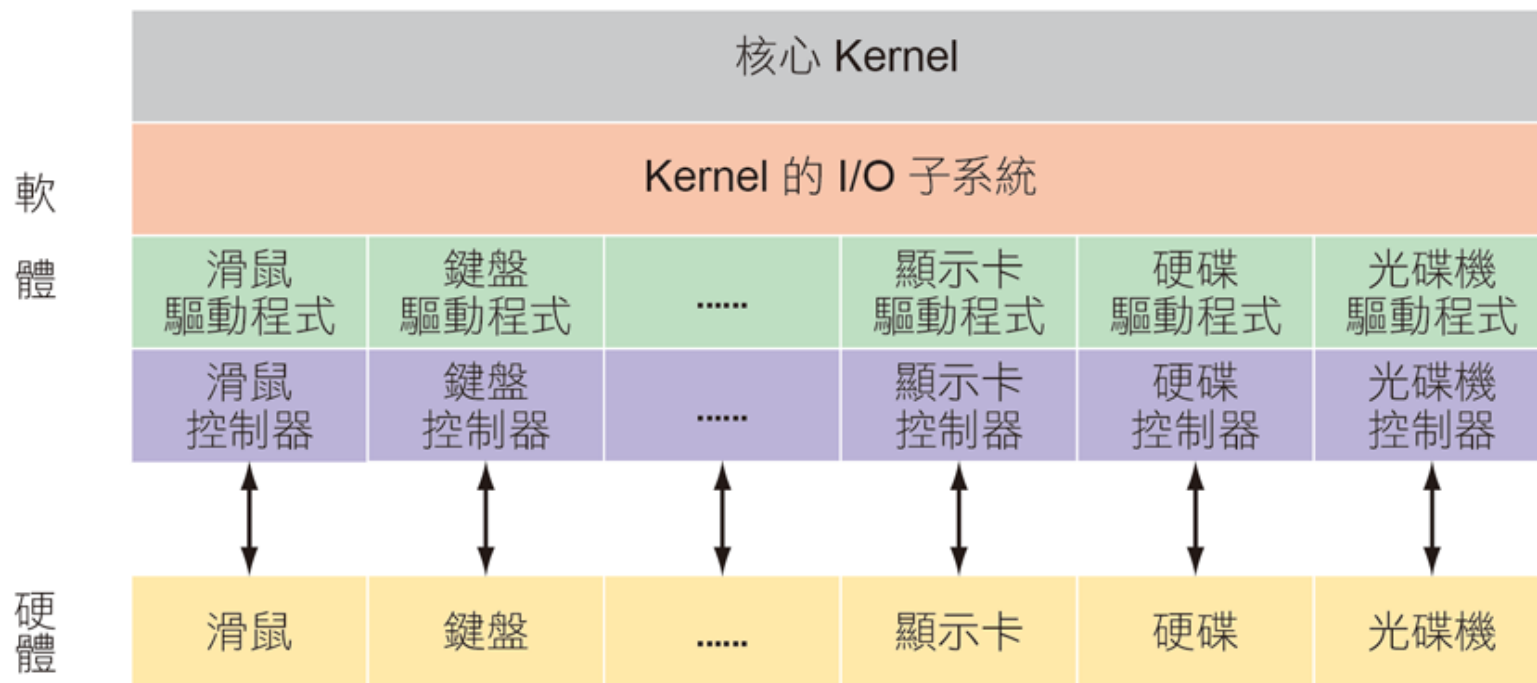
# 檔案的保護

- ◆ Windows 7 作業系統的檔案安全性，可依不同使用者，設定完全控制、修改、讀取和執行、讀取、寫入等五種使用權限，進行檔案保護



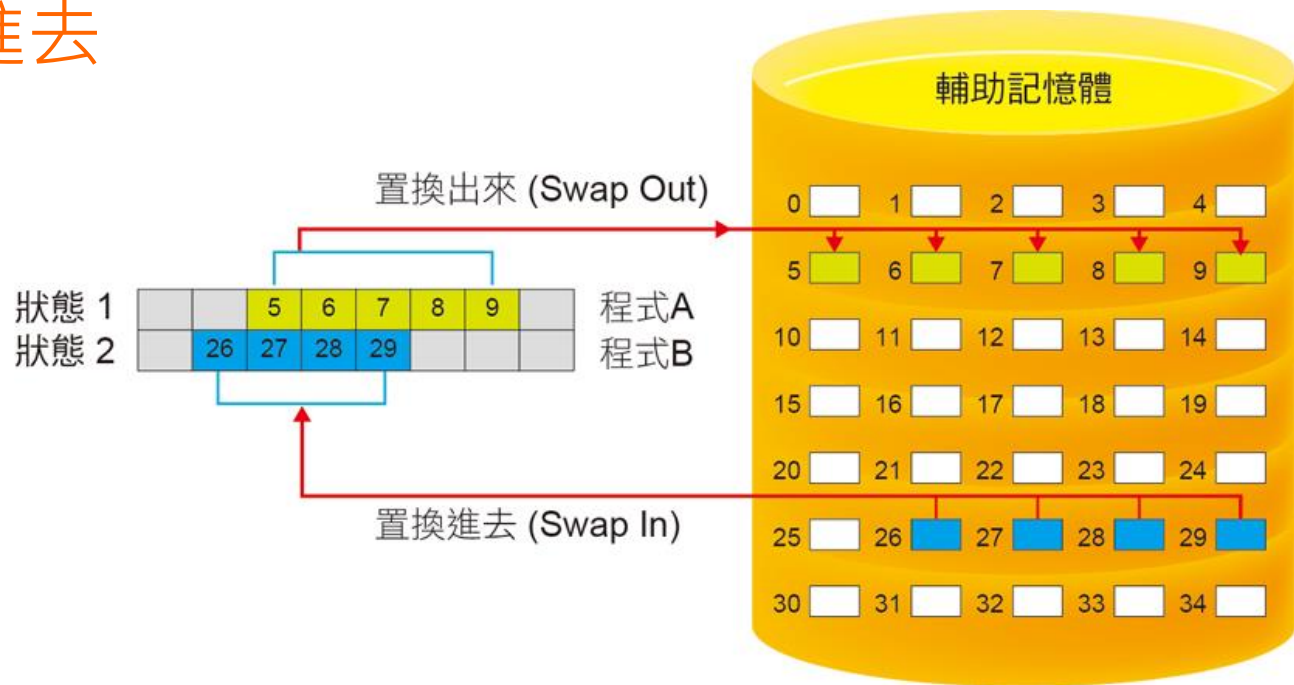
# I/O 管理

- ◆ 由 I/O 的驅動程式所組成，這些驅動程式能和 I/O 控制器溝通，完成應用程式所交付的任務
- ◆ 例如應用軟體播放 MP3 時，音效卡驅動程式會處理硬體播放的細節，方便軟體完成播放



# 記憶體管理

- ◆ 當使用記憶體的需求大於主記憶體時，OS 會在硬碟切割一塊區域，以分頁（**Pages**）的方式，將主記憶體中放置過久、較少執行的分頁先置換出去，存入磁碟中，再將需要立即執行的程序置換進去



# CPU 行程管理

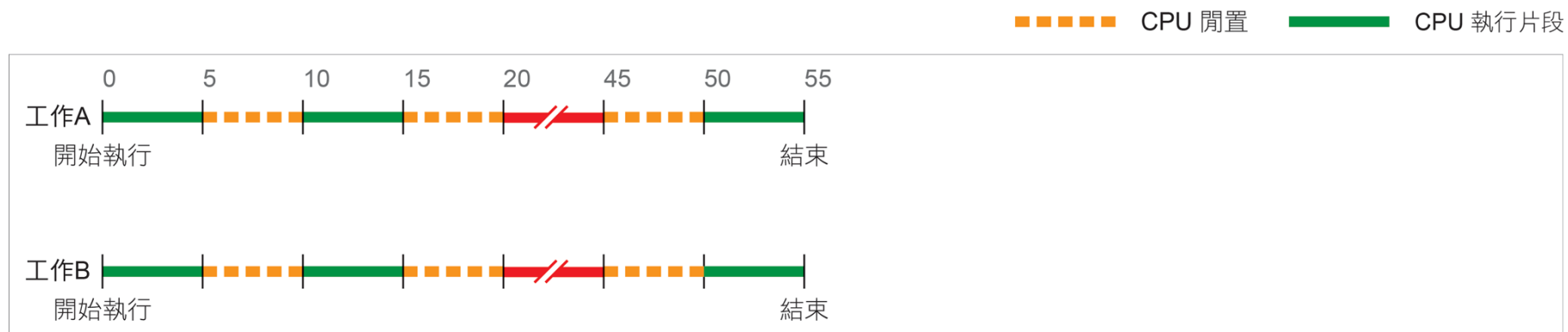
- ◆ 行程 Process

- ◆ 將儲存在輔助記憶體的程式載入系統執行

- ◆ 行程管理

- ◆ 同時執行多個行程時，要能妥善安排每個行程執行的順序

- ◆ 實例

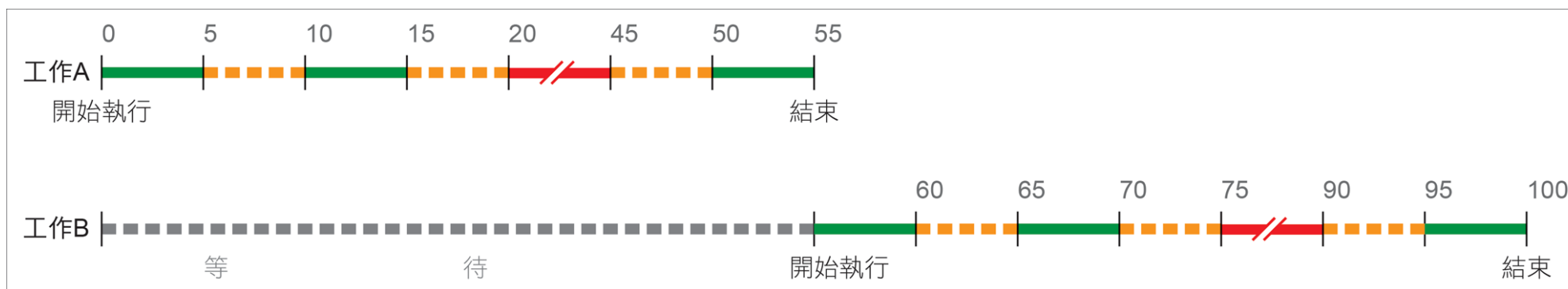


(A) 兩件待執行的工作

# 行程管理實例

- ◆ 沒有行程管理時

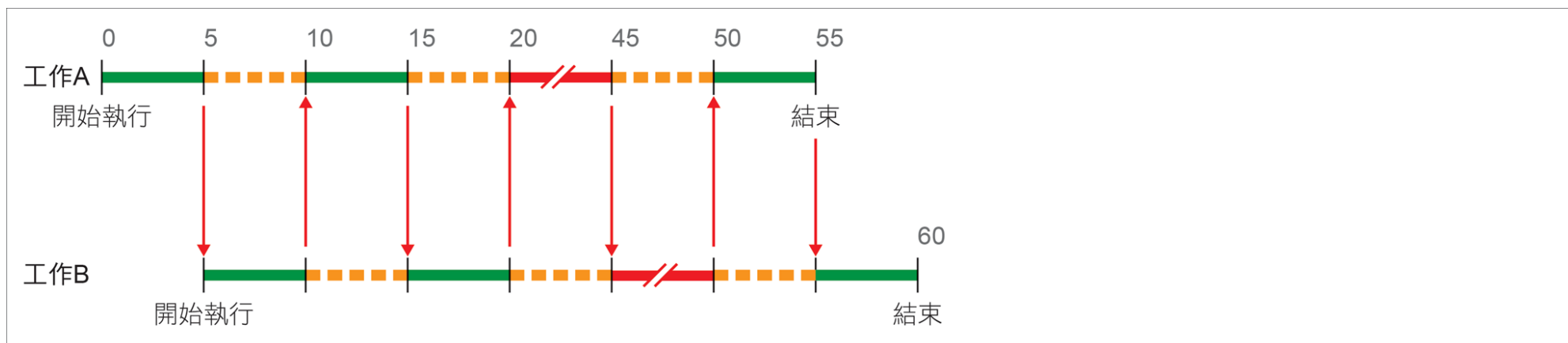
- ◆ 工作 B 必須等待工作 A 執行完後，才能開始執行，CPU 有許多時間都閒置，整體使用率只有約 50%



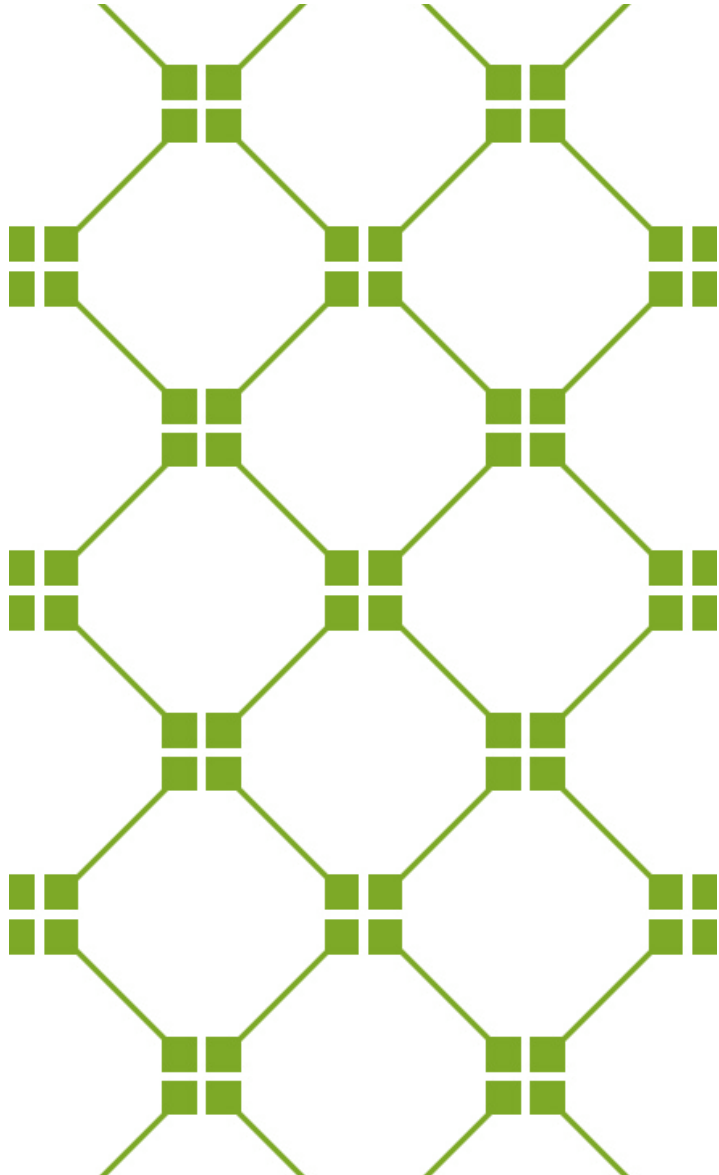
(B) 沒有行程管理

# 行程管理實例

- ◆ 有行程管理時
- ◆ 能充分使用CPU



(C) 有行程管理



See You  
Next Week :D

---