

# 系統程式期中報告

報告主題:虛擬機研究,編譯器研究,**WINODWS** 系統程式研究

資工二\_**110710538**\_李宗翰



## 目錄

什麼是虛擬機?-----	3.4
虛擬機的特點-----	4.5
虛擬機的時代背景-----	5
虛擬機的基本原理-----	5.6
什麼是編譯器?-----	7
編譯器的流程-----	7.8
編譯器的歷史-----	8.9

**什麼是 WINDOWS 系統程式?-----**

**9.10**

**WINDOWS 程式語言-----**

## **什麼是虛擬機?**

虛擬機器是行為類似實際電腦的電腦檔案 (通常稱作映像)。換言之，就是在電腦內建立一部電腦。它就像任何其他程式一樣在視

窗內執行，在虛擬機器上給予終端使用者的體驗就如同在主機作業系統本身一樣。虛擬機器自系統其餘部分隔離進行沙箱化，意味虛擬機器內的軟體無法自電腦本身逸出或竄改。如此便可製造出理想的環境，以使用來測試其它作業系統（包括 Beta 版）、存取感染病毒的資料、建立作業系統備份，以及在作業系統上執行原本未打算執行的軟體或應用程式。

同一台實體電腦上可同時執行多部虛擬機器。對伺服器而言，多個作業系統會並行執行，並由一個稱作 Hypervisor 的軟體加以管理，而桌面電腦通常會採用一個作業系統來執行其程式視窗內的其他作業系統。每個虛擬機器都會提供自身的虛擬硬體，包括 CPU、記憶體、硬碟、網路介面及其他裝置。虛擬硬體會對應到實體機器上的實際硬體，藉此減少對實體硬體系統的需求，達到節省其花費，以及其相關的維護費用，同時減少對電力及冷卻力的需求。

## 虛擬機的特點

虛擬機器的一個本質特點是執行在虛擬機器上的軟體被局限在虛擬機器提供的資源里，也就是說它不能超出虛擬世

界。「作業系統層虛擬化」不提供完整作業系統環境，將母機核心分給多個獨立空間的應用程式，不同於系統虛擬機需要運行完整作業系統，也不像程式虛擬機器運行特定程式語言。

## 虛擬機的時代背景

虛擬機出現在 1960 年代，為了提高珍貴的硬體資源（主機）利用率，促使企業對虛擬技術的研究與應用。而 IBM 在虛擬化方面的歷史由來已久，最早的虛擬化提出者和實施者就是 IBM。

1960 年代時，個人電腦 (Personal Computer) 尚未出現，只有政府與大型企業級的大型主機 (Mainframe Computer)，主要用於大量資料和關鍵專案的計算，例如銀行金融交易及資料處理、人口普查、企業資源規劃等。

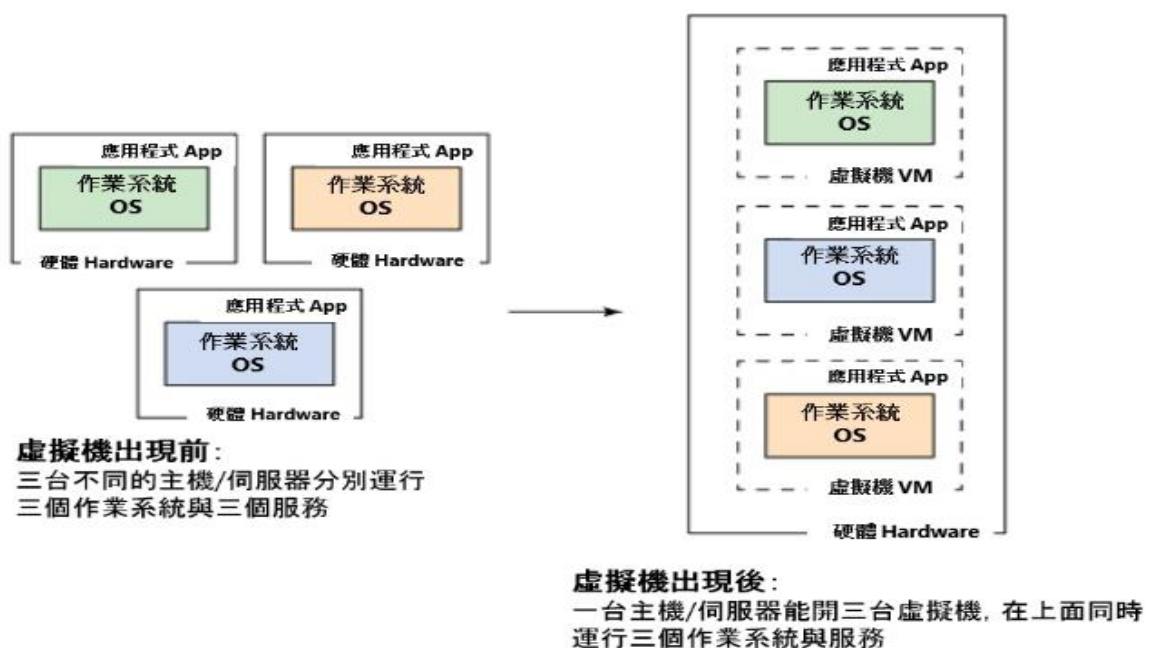
## 虛擬機的基本原理

根據「切割硬體資源進行彈性分配」與「安全的系統測試

環境」等需求的背景下，，IBM 開發出了「**虛擬機**」  
(Virtual Machine) 作為解方。

運用軟體技術、如 CPU 排程與虛擬記憶體，作業系統就能創造出一個多處理程式的假象：每個程式都具備自己的記憶體（雖然是虛擬的），且在屬於自己的處理器上面運作。

也就是藉由軟體虛擬技術，提供一個與底層硬體功能一模一樣的介面，這樣系統就能為每個程式都提供了一份底層硬體的 Copy，稱為**虛擬機 (Virtual Machine)**。簡單來講，就是用軟體技術模擬出一個硬體的環境。如圖下：



# 什麼是編譯器?

**編譯器**（compiler）是一種電腦程式，它會將某種程式語言寫成的原始碼（原始語言）轉換成另一種程式語言（目標語言）。

它主要的目的是將便於人編寫、閱讀、維護的進階電腦語言所寫作的原始碼程式，翻譯為電腦能解讀、執行的低階機器語言的程式，也就是執行檔。編譯器將原始程式（source program）作為輸入，翻譯產生使用目標語言（target language）的等價程式。原始碼一般為高階語言（High-level language），如 Pascal、C、C++、C#、Java 等，而目標語言則是組合語言或目標機器的目的碼（Object code），有時也稱作機器碼（Machine code）。

## 編譯器的流程

原始碼（source code）→ 預處理器（preprocessor）→ 編譯器（compiler）→ 組譯程式（assembler）→ 目的碼

( object code ) → 連結器 ( linker ) → 執行檔  
( executables )，最後打包好的檔案就可以給電腦去判讀  
執行了。

## 編譯器的歷史

早期的電腦軟體都是用組合語言直接編寫的，這種狀況持續了數年。當人們發現為不同類型的中央處理器（CPU）編寫可重用軟體的開銷要明顯高於編寫編譯器時，人們發明了高階程式語言。由於早期的電腦的記憶體很少，當大家實現編譯器時，遇到了許多技術難題。

大約在 20 世紀 50 年代末期，與機器無關的程式語言被首次提出。隨後，人們開發了幾種實驗性質的編譯器。第一個編譯器是由美國女性電腦科學家葛麗絲·霍普（Grace Murray Hopper）於 1952 年為 A-0 系統編寫的。但是 1957 年由任職於 IBM 的美國電腦科學家約翰·巴科斯（John Warner Backus）領導的 FORTRAN 則是第一個被實作出具備完整功能的編譯器。1960 年，COBOL 成為一種較早的能在多種架構下被編譯的語言。



高階語言在許多領域流行起來。由於新的程式語言支援的功能越來越多，電腦的架構越來越複雜，這使得編譯器也越來越複雜。

早期的編譯器是用組合語言編寫的。首個能編譯自己源程式的編譯器是在 1962 年由麻省理工學院的 Hart 和 Levin 製作的。從 20 世紀 70 年代起，實現能編譯自己源程式的編譯器變得越來越可行，不過還是用 Pascal 和 C 語言 來實現編譯器更加流行。製作某種語言的第一個能編譯器，要麼需要用其它語言來編寫，要麼就像 Hart 和 Levin 製作 Lisp 編譯器那樣，用直譯器來執行編譯器。

## 什麼是 WINDOWS 系統程式?

**Microsoft Windows**（中譯：視窗作業系統）是微軟公司以圖形化使用者介面為主推出的一系列專有商業軟體作業系統。其問世時間為 1985 年，起初為運行於 MS-DOS 之下的桌面環境，其後續版本逐漸發展成為主要為個人電腦和伺服器使用者設計的作業系統，並最終獲得了世界個人電腦作業系統的壟斷地位。此作業系統可以在幾種不同類

型的平台上執行，如個人電腦（PC）、移動裝置、伺服器（Server）和嵌入式系統等等，其中在個人電腦的領域套用內最為普遍。

## WINDOWS 程式語言

### 系統程式語言 (System Programming Language)

系統程式語言是用來撰寫和電腦硬體 (computer hardware) 溝通的軟體，常見的例子是驅動程式 (Windows driver)。初學者不會碰到這一塊，不需要費太多心思讀這類資料。常見的系統語言是 C 和 C++。

### 應用程式語言 (Application Programming Language)

應用程式語言是用來撰寫給使用者 (user) 使用的應用程式。根據使用者界面的不同，有可能是圖形界面程式、網頁程式、行動程式、命令列程式 (註) 等。通常會使用靜態型態語言 (statically-typed language) 來寫，因為動態型態語言 (dynamically-typed language) 用來寫中大型程式會較難維護。

## 通用型命令稿語言 (General-Purpose Scripting Language)

這類型語言多是指動態型態的通用型程式語言，直接以原始碼的形式發佈其程式。這些語言通常用來寫網頁程式和命令列程式。由於動態型態天生的侷限，這類型的語言較不適合寫中大型程式。

## 命令列腳本語言 (Command-Line Scripting Language)

命令列腳本語言是內附在命令列環境內的語言，沒有函式庫的概念，藉由呼叫命令列工具（即指令）來完成各項任務。命令列腳本會以原始碼的形態來散佈程式。這類語言會用來寫命令列程式和自動化腳本，而不會用來寫應用程式。

## 科學運算語言 (Scientific Computing Language)

將電腦語言用在科學方面的運算，像是物理 (physics)、工程 (engineering)、生物 (biology)、醫學 (medicine)、金融 (finance) 等領域的計算。有些語言會提供互動式運算環境，有些則是純批次運算。



