

台港及海外中文报刊资料专辑

827853

电子

500

创

23132

编

T·5

1

电讯

第 5 辑



工业

书目文献出版社

出版说明

由于我国“四化”建设和祖国统一事业的发展,广大科学研究人员,文化、教育工作者以及党、政有关领导机关,需要更多地了解台湾省、港澳地区的现状和学术动态。为此,本中心编辑《台港及海外中文报刊资料专辑》,委托书目文献出版社出版。

本专辑所收的资料,系按专题选编,照原报刊版面影印。对原报刊文章的内容和词句,一般不作改动(如有改动,当予注明),仅于每期编有目次,俾读者开卷即可明了本期所收的文章,以资查阅;必要时附“编后记”,对有关问题作必要的说明。

选材以是否具有学术研究和资料情报价值为标准。对于某些出于反动政治宣传目的,蓄意捏造、歪曲或进行人身攻击性的文章,以及渲染淫秽行为的文艺作品,概不收录。但由于社会制度和意识形态不同,有些作者所持的立场、观点、见解不免与我们迥异,甚至对立,或者出现某些带有诬蔑性的词句等等,对此,我们不急于置评,相信读者会予注意,能够鉴别。至于一些文中所言一九四九年以后之“我国”、“中华民国”、“中央”之类的文字,一望可知是指台湾省、国民党中央而言,不再一一注明,敬希读者阅读时注意。

为了统一装订规格,本专辑一律采取竖排版形式装订,对横排版亦按此形式处理,即封面倒装。

本专辑的编印,旨在为研究工作提供参考,限于内部发行。请各订阅单位和个人妥善保管,慎勿丢失。

——— 京 图书馆文献信息服务中心

软件开发技

中文综合

供专门应

整体系统

普通人的

软件专门

2001未来

书 号 ———
T-5

登记号 827853

杜忆民 7

23

74

76

应用程序

发酵程序的电脑化

统计软体 The Statistician功能强、应用广

Word Plus——一个容易使用的错字检查软件

文件名的简易加密和解密方法

杨 胜 17

倪安顺 34

48

张佛庆 50

图形软件

IMPRES 600 绘图编辑系统在工程上的应用

图形输入系统简介

谢匡海 52

邱文裕 57

(下转封三)

商业推广的最佳拍档——图形软件		60
电脑辅助绘图软体简介	陈荣中	66
其 他		
“动口不动手”的电脑操控模式		27
语言辨识系统的工作形式		29
微电脑环状通讯网路设计	杨耀波 刘昌焕	40
电脑科学基础之一——图论及其应用	绍 华	70

电 子 电 讯 工 业 (5)

——台港及海外中文报刊资料专辑 (1986)

北京图书馆文献信息服务中心剪辑

书目文献出版社出版

(北京市文津街七号)

北京百善印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

787×1092毫米 1/16开本 5 印张 128千字

1987年3月北京第1版 1987年3月北京第1次印刷

印数1—2,000册

统一书号: 15201·12 定价: 1.30元

〔内部发行〕

中文綜合軟件的先鋒「大五」



製作經年，耗費二千多萬新台幣的首套中文綜合軟件 Big-5，終於在六月下旬出來了。究竟這套軟件的功能如何與舶來品的分別如何，本文有詳細分析。

自 IBM PC 旋風橫掃世界市場之初，台灣部份有遠見的硬件廠商鑑於美國市場廣大，如能把握時機，當可在 PC 狂潮中取得一席之地，於是乎紛紛尋求與電子工業研究所技術合作，共同開發 IBM PC 相容產品。惟出人意料之外的是，首批銷往美國的電子所版本台灣十六位元私人電腦，竟因被控涉及仿冒而遭到擋關。其後雖因電子所的答辯，以及和 IBM 協商，修改了 BI-OS 後過關，但經此耽擱，在時機上已造成無可彌補的損失。

台灣業者受此挫折，紛紛企謀改善，利用各種方式來促銷十六位元私人電腦。同時，經濟部為輔導廠商進行研究發展，以加速推動資訊工業，特指示由資訊工業策進會負責協調，聯合七家硬件廠商及六家軟件公司，分包

趕工，共同開發適合台灣生產的十六位元電腦使用的五大套裝軟件 (BIG-5)。這是台灣資訊界有始以來規模頂大的一次合作案件。

誠然，BIG-5 的目標是為了解昇台灣軟件業者的開發能力，增進軟、硬件廠商間的合作關係。並希望經由軟硬件的配合，直接提昇台灣資訊產品行銷國際市場的競爭力。從八四年四月開始發展，其間雖因故一延再延，但總算在今年六月底宣佈 BIG-5 正式誕生。

基本上，BIG-5 包含了：
1. 文書處理；2. 檔案管理；3. 電子試算表；4. 商業繪圖；5. 通訊等五種主要功能，以 C 語言及組合語言作為發展之語言。在整合性功能上，由於當初考慮市場時機，以及參與開發的單位都是獨立個體，是以選擇最低的整合層

次——系列產品，也就是共用一個使用者介面的一群程式來進行整合，所以整合層次上，BIG-5 和 PFS 是相似的。

由於 BIG-5 的架構是一種系列產品，故看成為五個獨立作業的單元，當需要整合使用時，再透過共通的使用者介面來交換資料。所以在功能上和 LOTUS 1-2-3 根本是屬於不同層次的產品。故此將依其個別的特性，來和分別與之對應的 WORDS-TAR、DBASE II、MULTI-PLAN 等產品做說明、比較。

1. 文書處理：BIG-5 提供了二種文書處理系統，分別是具有中文能力的 CWP 和純英文的 WPS。如果我們拿 WPS 來和文書處理的銷售冠軍 WORD-STAR 做比較的話，我們將發現，在基本功能方面，從編輯、刪除以至列印效果，二者

提供的基本功能都是一樣的，唯一的差別只是在操作的方式不同，以及表現在使用者的可親性上。

在特殊功能方面，WORDS-TAR 具有連鎖印製、拼字檢查以及字典維護等三種特殊功能，WPS 則沒有類似功能。不過 W-PS 提供了目前非常流行的視窗功能，這是 WORDSTAR 所欠缺的。這種視窗功能，能在執行文書處理時，透過視窗選取其它檔案資料，來進行檔案的合併。

至於選擇處理文書與非文書作業方面，WORDSTAR 可立即依據選擇的項目是文書或非文書來執行，而 WPS 則一律需先經由文書處理的步驟，當所要處理的項目是非文書時，必須先經由轉換，剔除不必要的資料後才能達到目的。所以在處理效率上，WPS 稍嫌緩慢。不過 WPS 上提供有半圖形組合的支援，在設計圖表的美觀上，則是佔了優勢。

2. 檔案管理：BIG-5 中的檔案管理 FMS 系統，和國外著名的資料庫管理系統套裝軟件 D-BASE II 非常相似，都是一種關連式資料庫管理系統。

在檔案架構功能方面，二者都具有開檔、修改、拷貝、刪除以及列印等基本功能，不過 D-BASE II 上多了重新命名 (RE-NAME) 的功能。

在資料處理方面，則二者同樣都可執行增加、刪除、修改以及列印等工作。在資料型態上 D-BASE II 具有邏輯函數的檢查功能，FMS 則沒有。不過 F-MS 能同時開啓二個索引檔，而 D-BASE II 則僅能開啓一個索引檔。

在容量方面，每片磁片所能容納的檔案數目，D-BASE II 無所限制，FMS 則最多限制在二十二個檔案。D-BASE II 每個檔案可容納 65,500 筆記錄，F-MS 則到 65,500 筆記錄。D-BASE II 每筆記錄可分三十二個欄位，FMS 則可多到六十個欄位。D-BASE II 每個欄位最多可放 254 個字元，FMS 則可放到 255 個字位。

在檔案安全性方面，FMS 提供有密碼控制方式，D-BASE II 則沒有。至於操作方式，D-BASE II 是經由使用者下命令驅動，而 FMS 則是以指引方式驅動。所以 D-BASE II 較 FMS 在使用上更具有彈性。

3. 電子試算表：基本上 BIG-5 中的電子試算表和 MULTIPLAN 非常近似。熟悉 MULTIPLAN 的使用者將在 BIG-5 中看到很多熟悉的影子。

無論是儲存格的格式，運算子以及使用單一文字的指令集，BIG-5 和 MULTIPLAN 都是非常類似的。

這二者最顯著的差別，就是 BIG-5 的電子試算表具有中文能力，而這項特點就足以增加它在台灣市場的使用率。

4. 商業繪圖：BIG-5 的 GRAPHIC 雖已開發出來，但操作手冊則尚未撰寫完成。商業繪圖系統的功能在於提供決策顯明易讀的圖表，因此要比較 BIG-5 和 LOTUS 1-2-3 的商業繪圖能力要從圖型顯現能力上作比較。

在圖形表現方式上，二者都能顯示長條圖、堆疊長條圖、線形圖以及扇形圖。而 LOTUS 1-2-3 尚能顯示散佈圖，BIG-5 上的 GRAPHIC 則無法。

在圖形的標題、標記以及比例尺功能方面，二者都功能齊備。而 BIG-5 更能視螢幕大小，自動控制附加說明字體的大小。LOTUS 1-2-3 則須由使用者來自訂圖解說明。

至於繪圖圖型資料的來源，LOTUS 1-2-3 的繪圖功能是包容在試算表中，可直接經由試算表中的資料產生圖形。當須經由印表機列印圖形時，則透過資料交換格式來儲存列印。而 BIG-5 中的繪圖功能對於外來的資料則需先透過共通的 CSV 檔案來傳遞資料，再做顯示列印。5. 通訊：BIG-5 爲了使台灣生產的十六位元私人電腦能與其它電腦連線以共享資料，故採用非同步通訊方式，經由 RS-232 來和大型電腦，或其它個人電腦連線。且透過此一功能，還能把私人電腦模擬成基本型終端機使用。此外，爲了加強對使用者的親和性，BIG-5 亦用指引方式來誘導使用者，依需要選擇各種功能。

從市場的導向來看，整合性套裝軟件是市場上的新趨勢，然而透過共用一個使用者介面來進行整合，層次上就比利用一套多功能程式去達成整合效果來的差。在講求速度及便於使用的今日，以 BIG-5 仍算脆弱的體質，要在歐美市場，打出一片天下來，實在是個疑問。

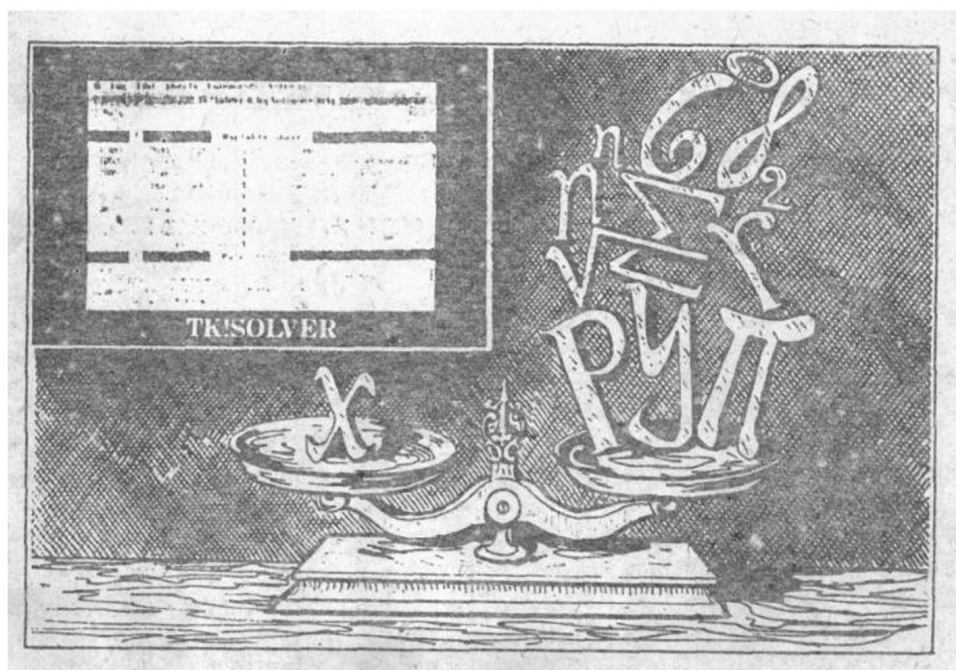
不過 BIG-5 在某些方面仍有顯著性的成功，它畢竟提昇了台灣幾家軟件公司的開發能力，而這些技術能力都是台灣資訊界無比珍貴的資源。

同時，更重要的事，BIG-5 具有處理中文的能力，這是任何舶來品無法應及的。

ME

(原載：現代電子[港] 1985 年 10 期 138 --- 139 頁)

供 專 門 應 用 的 軟 件



現在已有很多微型電腦在市場上銷售，並且成為解決一些難題的重要工具。而這些問題的中心點有數端，如詞語的運用、資料的處理、或是財政會計上的分析。雖然現在的機械和電腦軟件確實是有相當的成就，但是甚麼才是較為專門化或技術上的應用呢？例如具有可表現其神奇的計算、發展至老練的統計分析，或是找著那些主觀的判斷。現在，大多數的情況是那些典型的電腦軟件批發商人，他們將會繼續把那些電腦軟件售給其客人。但是，他們所面對的困難就是那些標準系統已成為了受歡迎的包裝貨品，正如 LOTUS'S 1-2-3 和 AS-HION - TATE'S dBASE 一樣。但他們所做出來的結果，却會使人覺得非常失望。

可是，你又不能真正怪責那些批發商人，他們雖然擁有不斷增加數目及多元化的貨品，而這些貨品是完全受他們所操縱的。但事實上，今日的商业電腦軟件已大鬆運用包裝來吸引顧客，而包裝設計使商品變為與眾不同；而這也可能是技術上的需要

。這些經特殊包裝的電腦產品，可能是很難在市面上找到的。雖然如此，因很多牌子都是較為新及未為人所注意，所以他們的市場潛力較細。他們就不像那些一般用途的電腦軟件一樣，花費大量的金錢在廣告宣傳或刊物宣傳。

在這篇文章裏，我們希望給讀者們一個對電腦軟件產品如標準文字處理、資料庫管理和試算表系統等有一個認識。自開始以來，我們便要承認我們所討論的問題是有點特殊的應用，所以責任是應由你怎樣去定義你的需要。雖然，每一個人都知道，文字處理使大多數的人獲得好處，只要你能夠決定方程式處理器或最好的專門系統便可以解決你的需要。

我們現在可探討 4 種特別的「難題解決」的電腦軟件。首先是運算及代數套裝軟件，其設計是用來分解方程式或用來作為數學代用符號操作。第二是用核心硬件的統計系統。第三是專門或以技術為其基的系統。最後是作決定的輔助工具。而這輔助可以用來支持其主觀的判斷。

運算套件

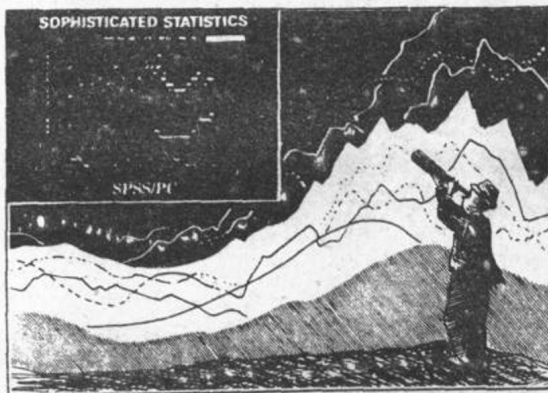
在那些會計師及財政分析家們的眼中，試算表是一件很有用的工具。但是，若我們去問那些數學家、會計師、或資料調查員有關這些紙的意見，你會發現到他們的意見有相當的保留。其問題是涉及多過一個的變數，而這變數是不能符合那些試算表的標準。因為他們只有固定的齒輪，而這些齒輪只是有同一方向。所以在分析問題時，是不斷重複的，要在同一變數裏有多次分解。同樣，這些工作程序是需要運用代數公式裡的數學符號操作，但這操作已超過試算表的容納量。

有一些特別裝配的電腦軟件，可以用來幫助我們處理日常事務的，而 SOFTWARE ARTS INC. 所生產的 TK! SOLVER 已被視為一個「方程式採掘器」，而這個系統是忠於那些會計師、化學家，或一般的商業界的人士。讓我們來看一個很簡單的例子，使我們能了解到 TK! SOLVER 怎樣去工作。

在視覺上，物件影像的大小及其位置是從視網膜上產生的，而這種現象是透過一個方程式表達出來，這方程式就是 $1/p + 1/q = 1/f$ 。而 p 是物體與視網膜間之距離； q 是影像與視網膜之間的距離； f 是物體的影像在視網膜上的長度。你可以建立一試算表的模型，而這模型是可以重複地從 p 或 f 的數值中找到 q ，但這亦同時可以找到 q 的。當你從方程式輸入 TK! SOLVER 的 RULE SHEET 後，你便有機會去在程序上的 VARIABLE SHEET 中特定的兩個變數。同時，電腦軟件將會提供並未選定變數的數值。

這個例子是非常簡單，但是它一定要說明 TK! SOLVER 與試算表計算的基本分別。這程序會表現出較佳的計算：你能把一系列的方程式輸入 TK! SOLVER 的話，TK! SOLVER 便會運用其 ITERATIVE SOLVE 方式，而這可幫助那標準的 NEWTON-RAPHSON 方程式來改進其估計，直至這估計為一最接近數值的可能。TK! SOLVER 同樣擁有一系列的價值作為任何變數，作為單位上的變換、繪圖或做表，都是以其所找到的作為基礎，然後便把它的結果印出來。同樣地，特別用途的 SOLVEPACKS 提供了模板，使其可提供適合的方程式、函數及一些量度某一個範圍內的單位，正如機械工程所用的或是財政計算時所用的量度單位。

但有一些事情，TK! SOLVER 是不會做的，就是組合及簡化幾何公式，因為它的輸出是限於數目字符號的操作。符號操作（正如用其他的基本例子，就是 $a + a = 2a$ ）是其範圍叫符號運算套件。或者，我們最好是稱它為 muMATH。muMATH 是由 DR. DAVID STOUTENMYER 所發展的，並且由 MICROSOFT CORP. 在市場



推廣，如果 TK! SOLVER 操縱很多方程式的話，我們可以說是 muMATH 操縱它們。固這是使它們由符號轉化為微分，多項式、及找出其導數和積分。

大型電腦的「順水而下」的影響是在工作場所裏運用數學性的電腦軟件，同時我們可以看到這幾年來，微型電腦的功能已不斷的增加及推高，在劍橋和麻省有一間 SYMBOLICS INC. 是由 DR. RICHARD PAVELLE 所開的，（巨型電腦裝置的市場調查員）覺得我們將會很快便看到那些試算表是包括代數功能的。而這些試算表將會比我們在今日所知道的更有動力及更有彈性的。

雖然 TK! SOLVER 及 muMATH 已面世了兩年多的時間，但它們基本上是依附那些小的市場，例如工程師、數學家、學校教育工作者及其它專門人士。但是，正如我之指出，它的潛能可以應用在很廣大的範圍問題上，而這些問題，亦是在不同的範圍裏的：如果你發現試算表不能給與你有彈性，而這正當你需要分析數字或數學上的關係時，這種電腦設備確實已減低你對電腦的注意了。

統計方法

很多市場調查員、社會科學家、政府及工業的行政計劃人員及一些調查人員都面對一個困難，這困難便是怎樣從眾多的數目字的資料來劃出一個統計的結論，而這種做法，却往往使人失望（失望是對試算表的失望）。舉一個例，LOTUS 1-2-3 的輸入可供作最低、最高、平均及標準的統計運算，但是它不能給與你一個較方便的方法去處理倍數運算、交叉列表及其它有函數。如果你雖要好像以上所說的功用，你一定要考慮一個套裝軟件 SPSS / PC。

SPSS / PC 是由一間名 SPSS INC. 所生產的，其性能可說是微型電腦中最好的，可說是社會

科學的統計程式，而這一座巨大的電腦統計機可以在很多大學的資料中心裏找到。並且，它已普遍地被那些調查員及專門分析家在多方面加以利用了。正因為微型電腦可保持主機電腦的做句形式及指令結構，因此用家可以將檔案在兩者間互相傳送。而這些版本的分別，主要是 SPSS 有其互相影響的模式，所以指令能夠在一個很短的時間內完成。但是巨型電腦版本的 SPSS 只可以在同時會集方式下操作才能發揮其效用。

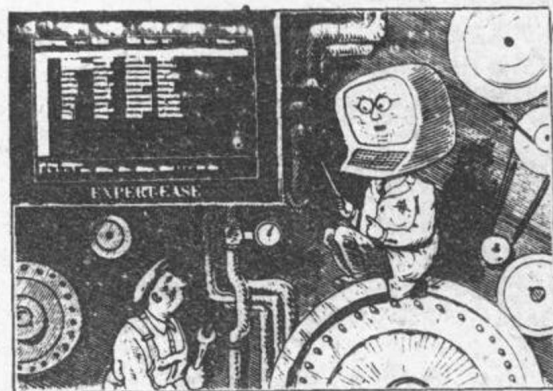
另外，有關此系統的統計功用，SPSS / PC 是包括 KERMIT 的檔案傳輸程式，所以你可以很容易地把那些資料在 SPSS 的版本當中傳輸。SPSS / PC 是可以輸出或輸入任何標準 ASCII 檔案的，使你可以在另一個程序當中得到寶貴的資料，而這個程序是貯在八塊磁碟。但是它明顯已知道一定要使用硬磁碟才可以做出來。

從經驗中學習

有一些技術問題是很明顯已減低或不能與數字一同工作的，假如，沒有一些方程式或統計分析會告訴你究竟機器中有什麼問題會發生的。而這個問題，可能是整部機械是完全損壞；或好像一個病人，究竟他為什麼會痛一樣，其痛苦是否因呼吸困難而令他窒息。這些要視乎性質而下的判斷，將會由那些專家們所決定，而他們的決定大多是基於他們的年積下來的經驗。但事實上，那些專家們都很难去獲得那些大量的調查資料，而且這是一個很昂貴的途徑去發展那「專家系統」，於是電腦程序設計便是專家們的意見及忠告，可以落到那些面對困難的人身上，於是典型需求擴大及經驗已專門化了。

直至最近，這種系統的價錢不單很昂貴，而且只有在大型電腦中心才使用它。雖然如此，但是去年 EXPERT - EASE 這一專門系統已開始達至微型電腦操作者的要求了。EXPERT - EASE 是由 DR. DONALD MICHIE 在蘇格蘭發展的 EXPERT - EASE 可使你成立一較夾雜的專門系統而這系統是特別在某些情形下所專用的，當你多次依賴同一的經驗根據時，運用 EXPERT - EASE 的第一部就是把專門的知識輸入電腦內，而這系統的資料全根據你的提供。因此，第一是你必要輸入其屬性，而這屬性是可成為一些分數，並可在程序中向你發問。第二，你一定要提供一些例句，實際上是教導電腦系統怎樣去設法取得答案。當你把所提供的屬性及例子輸入 EXPERT - EASE 後，它便可產生出一系列的法則，可使這些法則隨時應用。因此，此程序可以表示一系列的問題。

當你剛提供那些基本的專門知識及準備使那系統工作時，便開始有互相影響。而那 EXPERT -



EASE 是解答其所問的問題。但是這專門系統並不會用多過你所提供的屬性。反之，它是用一些由你輸入電腦的規條，而這些規條可使你找到問題是屬於那一個範圍，並且問題是怎樣排列出來讓你給他解答。你解答每一順序問題的方法，是由你從一系列答案中選擇一個可能答案。最後，當你已可解答所有問題後，EXPERT - EASE 會給你一些提示。實際上來說，那些人類專家所提供給電腦的屬性及例子，並不是好像一些人一樣，去在電腦系統當中尋求指導的。

EXPERT EASE 是一個極富彈性的工具，有了它的幫助，你可以發展一系統來幫助你在任何的困難中獲得解決方法。因為這系統已有一系列的規則。無論是醫學上的診斷、化學上的分析、工程設計圖側，或甚至是在甚麼適當時候選購那一隻股票；都可使用這一個系統。

同樣，那些數字的分析是決定於我們的應用是什麼。而這就是我們或許叫主觀判斷。其它的資料調查員都依賴其直覺所見的種類，而這些類別是透過進行實驗性試驗得來。

但在這個過程中，時常會有一些懷疑，就是因為所有的適當事實並不是透視的，同時，當一個決定走出來時是不正確的話，這是很困難去明白究竟什麼做錯了，及怎樣去避免將來有同樣的錯誤。雖然如此，但是新決定處理及支持的軟件，將會成功地提供一個主觀的決定。

EXPERT CHOICE 是由 DECISION SUPPORT SOFTWARE INC. 所發展的，它是其中一個最有能力的決定管理的電腦系統。這程序要你輸入所有有關連的狀況來運作出一個決定。然後便衡量每一個可能性。由它所輸入的來看，其程序組織了一些困難，這些困難是傳統從末為尖的結構問題，你的目標是主幹。因此，基本標準影響其目標是其大幹，較不重要的小支；如此類推，通常這程序可有六個等級。

舉一個例子，是那類困難適合 EXPECT CHOICE 的設法的。首先，讓我們考慮一個徒勞無功的工作，就是去決定那裏可容納一個有毒的廢

物收集站有很多的因素牽涉到決定，而這些決定當中又是客觀及有限制的。以地理學來說，接近地下水供應之地面、土壤種類等。但其它的因素却是極度主觀，例如，審美對鄰近地方的影響力、及對本地政治的反應等。

EXPERT CHOICE 可給與你一個方法去平衡主觀因素裏的客觀因素。只要你輸入客觀資料，並同時輸入主觀的關係入電腦程序內；這電腦便會衡量每一個因素，而這些因素是涉及你的直覺決定的，若使用「分析架構」程序，EXPERT CHOICE 將可提供一個答案，而這個答案是基於其並列的所有貼切資料，及你所提供的每一個因素的重要性。

當然，你仍然會去決定那些有關於主觀選擇的重要性，而這重要性是圍繞著這決定的。但 EXPERT CHOICE 可令他更容易去建立信心。同樣，有一個相同的能力，這程序須可使你去衡量每一個因素。

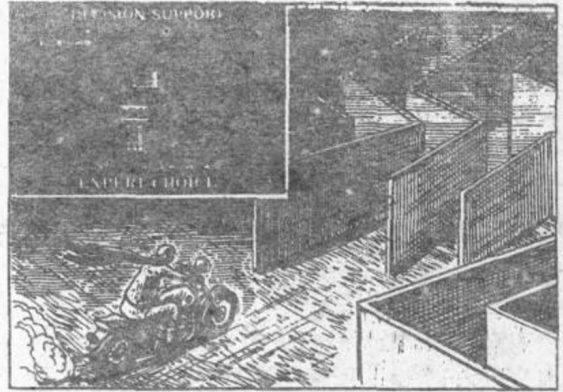
舉一個例來說，你想看到有什麼發生，而你想提供美學對衝的高度重要性，你便很容易地整理其比重，而這比重是關於推行及直接觀看其轉變怎樣影響程序。同樣，如果一個新的因素突然拿來應用的話，你便可以衡量其重要性。當你把这些輸入其程序時，你便可衡量其重要性。同時，你亦可獲取其修訂的資料，而這種情形是沒有重新思想整個困難。所以一個程序文件是一完整的決定程序，而你能夠利用這模型去指導你將來的工作時，可作出一個同樣的決定。

其實 EXPERT CHOICE 及 EXPERT EASE 都是質勝於量的，其相差的只是在表達上有分別。EXPERT-EASE 製造事實或知會一些專家的判斷，用者會被那系統發問一些問題，而用者亦可能在最後決定時，根本沒有一些基本的真確資料。另一方面，EXPERT CHOICE 是一個互相影響很強的系統，而這系統是十分倚靠用者的主觀判斷的。

支持決定的電腦軟件正是一門增長得很快的工作。除了 EXPERT CHOICE 也許是最好的電腦裝配外，一些同類型的電腦軟件也由 LIGHT-YEAR INC., ANALYTICA, HUMAN EDGE 和其它的公司一同於市面上發售。雖然有些人有一個很複雜而又主觀的決定。但現在他們都希望電腦支持那些吟吟不樂者的言語，使他們可以有機會享受一會兒。

其它解決問題的軟件

有四種「能解決問題」的軟件，而我們已看到



它是代表了明智及清晰的選擇。因為它們是廣泛地在程序上運用的。正如那些受歡迎的試算表及數據庫管理一樣。另外一個值得提供的方法，就是那些混雜的東西，就是 VERTICAL MARKET 的產品，它提高那些特別的模板、模型或檔案來加強以模板為基的系統，而這些都是可以定義為工作上特殊應用。

我們剛提出過 SOLVERPACKS 是可加強 TK! SOLVER 的能力。同時，一類相似的產品已隨着 1-2-3、dBASE II、VISICALC、MULTIPLAN 等著名軟件而製造出來。舉一個例，如果要把試算表推銷給那些稅務會計、實業管理、投資分析或一些在數字上作特殊用途的人，你可以找到附加產品，而這種產品便可以很有效的去解決你的需要。

在結尾時，我們一定要再次強調在開始之時所作的論點。這篇文章不能詳細供應「解決問題」電腦軟件的資料。但是，我們希望提供你們一些參考。因為每一個人的需要都有少許的不同，而且所有選擇最後是應由你做的。現有一些提示給你們，就是找尋電腦軟件時，要考慮到：

- (一)明確地規範你的需要，你不能期望任何人去為你定義。假若你沒有一個定義的話，就算它已跳進你的衣袋裏，你也不覺得它是最適合你的。
- (二)利用多方面的資料——不單是那些本地的代理商——來得到一些有關其發售的資料。或許，其中一個就是你所需要的。但如果你不能肯定的話，你亦可以信賴那些在市面上流行的電腦軟件介紹錄、一些專業團體、或是那些專業諮詢人。
- (三)透過不斷觀察，現今有很多的電腦軟件推廣者認識到，這個市場只有小小的銷路給與新的試算表或其它標準的應用。他們會大大增加軟件的專門性，使其成效能在那些特殊的產品上，發揮其效能出來。

現在，微型電腦及軟件確實是可以幫助你去解決困難，無論它是否精於運算或造決定，只要少許探索及嘗試便會發掘出其不可限量的潛能。

整體系統發展工具

杜憶民

一、前言

微電腦目前在國內已經非常風行，不論是八位元或十六位元微電腦，均有甚多發展系統及應用系統在市場上出現。由於微電腦在開始發展之時，其功能較差，所能運用的範圍較窄小，因此並沒有人對於系統發展流程及工具加以注意，因此其生產力甚差，尤其在軟體系統發展上更為明顯。

我們如果拿微電腦的軟體發展工具和小型電腦系統中的發展工具相比較的話，會發現微電腦發展工具實在是少得可憐。但是由於微電腦之體積甚小，而且硬體發展的新技術更是日新月異，在短短的十年之中風行全球，現在每一個微電腦均擁有相當可觀的處理能力，加之分散式系統體系已漸成局面，許多巨大的系統均是由許多微電腦組成，因此微電腦在應用系統中的地位更是日形重要。

雖然微電腦的硬體技術發展速度驚人，可是軟體技術却步履緩慢，無法匹配硬體技術的發展速度。由於軟體發展及維護費用不斷提高，已成為系統成本之最大項目（見圖 1-1），因此微電腦業者及研究人員也開始採用許多方法來提高軟體生產力，降低成本，縮短發展時間，同時也提高軟體品質。

在這些逐漸發展出來的方法中，包括：

1. 軟體工程及方法（Software Engineering & Methodology）。
2. 結構化設計、規劃、及程式編寫。
3. 軟體元件化。

4. 改善系統發展環境。

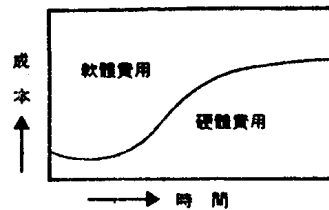


圖 1-1 系統成本發展趨勢圖

以上所列舉的僅為舉世大者，到目前為止這方面的發展仍然不斷進行，將會有更新方法或新理論不斷出現，以使軟體發展步入軌道。

本文所要討論的內容就是「如何改善系統發展環境」，在文中並不討論任何不切實際的理論，僅只略試由實用的觀點（Pragmatic Approach）來說明如何建立一個較有效率的系統發展環境，以大幅度提高軟體生產力。在文中所談的工具均已在電子工業研究所建立，而且經測試後發現效果良好，值得推廣。

二、系統發展環境（System Development Environment）

在傳統的觀念上，一個軟體系統必須要在未來程式所處的系統上發展，例如在 S / 370 上發展一個程式，而這個程式將來也必然在 S / 370 中運作。這種觀念在小型以上電腦可以解釋得過去，因為小型以上電腦已具備有相當的軟硬體能力，可以一

方面供人發展程式，同時另一方面可以進行日常的作業程序。

但是在微電腦系統中，這種狀況就稍有不同了。由於在目前這個階段中，微電腦的能力還不夠，尤其是軟體力量不足，不能提供使用人員予強而有力的支援，因此系統發展往往是事倍功半，得不償失。舉一個最簡單的例子，如果使用 Intel MDS 發展系統，則每次僅能提供一個單使用者 (Single User) 的發展環境，若有一個人使用 MDS 來鍵入程式的話，則其他人就只能在一旁乾瞪眼，聽任一些有用的偵錯工具，例如 ICE 閒置無用，徒呼奈何。

要避免這個問題有一個簡單的方法，就是將系統發展環境 (Development Environment) 和運作環境 (Operating Environment) 分離開來，以求在發展環境中發揮最大的生產力。所謂發展環境就是指發展程式的軟體設施，而運作環境指的則是程式發展後所載入 (Load) 的軟體操作設施。(見圖 2-1)

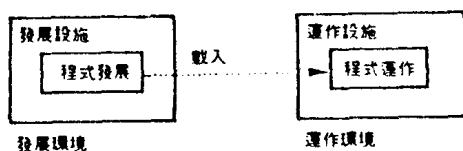


圖2-1 發展環境及運作環境

爲了提高微電腦軟體的生產力，我們可以把發展環境及運作環境分離開來，而使發展環境的能力大爲提高，可使用小型電腦以上的機器作爲發展設施來發展微電腦程式，目前所見常用的發展環境有 PDP-11, VAX-11, DEC-10, NOVA, IBM S/370 等機器，而運作設施則包括 6502, 8080/8085, Z80, Z8000, 8086/8088, 68000 等常見之微電腦元件組合。

使用小型以上電腦作爲發展系統可有下列數個優點：

1. 使用這些小型以上電腦之多使用者作業系統 (Multiuser Operating System) 可使許多使用者共用同一部發展系統，不致於發生一些簡單的工作 (例如 Editing) 佔據了整個發展系統，浪費了寶貴的系統資源。
2. 由於小型以上電腦系統可提供強而有力的系統發展支援，使用的人能使用方便而有效的功能來進行系統發展工作，例如極好用的編輯程式，多種樣式的檔案處理功能，以及較爲友善的人機介面等，這些都能大大提高系統生產力。
3. 由於小型以上電腦軟硬體的功能較微電腦強大得多，因此所有工作都能在很短的時間內做完，不但提高生產效率，縮短發展時間，而且對於程式發展人員因等待而產生的沮喪程度大爲減小 (見圖 2-2)。

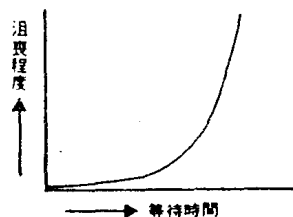


圖2-2 等待沮喪曲線圖

4. 小型以上電腦大部份均能提供良善的檔案管理功能，因此各種資料可以儲存在中央資料庫之中，工作人員可以很輕易使用其他人已發展出來的各種程式及工具，強化分工及合作的功能，同時對於資料的保護與管理均比儲存於微電腦中要有效得多。同時由於小型以上電腦均提供較大之磁碟，因此資料之儲存也比較方便。這種集中式的資料管理，若能配合良好的發展制度，在中型以上的計劃中，將會獲得極大的利益。同時由於資料存取方便，使得發展軟體元件 (Software Components) 的環境能很輕易的設立起來，大幅提高軟體生產力及品質。
5. 使用小型以上電腦做爲發展系統較易於統一發展

環境，而導致發生設計人員須學習多種作業系統的現象。這種現象在快速發展的公司中最容易發生，另外在使用多種以上微處理器的公司中也常見到。這種狀況對於人力運用是一種極大的浪費，而且所發展出來的產品又往往不具有共通性，亦造成資料資源上的損失以及日後維護上的困難，應該極力避免才是。

6. 許多公司原來即已擁有小型以上電腦，若能在這些電腦上建立發展系統，則所花費的代價將遠比採購多部微電腦發展系統要來得便宜得多，甚至於為此採購一部小型電腦往往也能在成本上獲取利益。

三、發展程序 (Development Procedures)

在本文所欲提出的是一個已經建立好的發展環境，這個架構並非改善發展環境的惟一解答，最重要的是系統架構的觀念，因為只有在這種統一的架構之下，軟體的生產力才能作最大幅度的提昇。

在這個架構中，事實上包含了二個發展環境：

1. 小型以上電腦發展環境。
2. 微電腦發展環境。

整個架構可由圖 3-1 表現得很清楚。

在這個架構中，小型以上電腦發展環境是最主要的工作環境，因為大約百分之八十五以上的活動都在這個環境中施行，而微電腦發展環境則主要作為整體測試以及介面測試時之用。

在本節中，我們就以圖 3-1 作為藍圖來探討一下軟體系統的基本發展程序，在以下的說明中，括弧內的數字即用以對應圖 3-1 中各元件左上角的數字。

1. 建檔

當一個設計者想要發展一套程式時，他必須先進行規劃、設計、及程式寫作等工作，這些部份因不在本文說明範圍之內，故不予討論。當這些工作做完之後，必然會產生下述二種文件：程式手稿及各種相關文件，這時所要做的第一件事情就是將程式及文件內容鍵入電腦之中，以便遂行隨後而至的

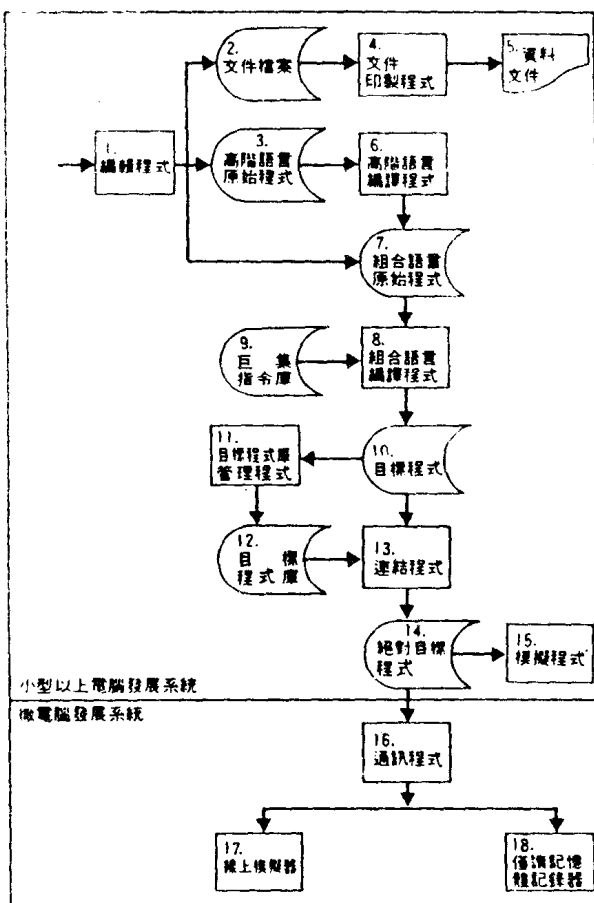


圖 3-1 發展環境架構圖

各項發展程序，這時所必須使用的工具就是編輯程式 (Editor 1)，這個工具可協助打字人員迅速的將各種資料輸入電腦之中，建立起相對應的檔案。文件資料會被放入資料檔案中 (Document Source 2)，高階語言程式則被放入高階語言程式檔案中 (High-Level Language Source 3)，而組合語言程式則被放入組合語言檔案中 (Assembly Language Source 7)。

2. 製作文件

將文件內容儲存於電腦之中有下述幾個好處：

- (1) 易於保存及管理。
- (2) 易於修改。
- (3) 易於印製。

而文件印製程式 (Formatter 4) 的最主要功能就在於印製漂亮整齊的資料文件，同時還能具有編輯、編頁、分段、調整等功能，與人工作業相較之下，其能力真不可同日而語。在此順便值得提起的，就是目前有關於中文資料之文件印製程式仍有待我們電腦業者共同努力，因為印製及編排中文資料等工作，實在最適合電腦來執行，投入大量人力在這類工作上，真是一種巨大的人力浪費。

3. 高階語言程式編譯

高階語言程式(3)可由高階語言編譯程式(Cross-Compiler 6)負責轉譯成組合語言程式(7)，因此高階語言程式經過此一階段之轉譯之後，即和組合語言程式沒有二樣(在此處之方法僅為目前許多公司所採行的方法，事實上並非必定要轉譯成組合語言程式，這點在本文中並不多加論述)。若欲提高軟體生產力，大量使用高階語言是一項極重要的方法，因為這樣可以大量減少發展所需時間，降低程式複雜度，提高可靠度，並減少維護時之困擾。

4. 組合語言程式編譯

組合語言程式(7)必須經過組合語言編譯程式(Cross-Assembler 8)將程式轉變為目標程式(Relocatable Object 10)，以便進行組合。在此一編譯過程之中，編譯程式可能自巨集指令庫(Macro Library 9)檢取所需的巨集指令加到原始程式中參加編譯，因此巨集指令庫是第一個可以用來儲存軟體元件的處所。

5. 程式組合

當所有目標程式00都已經產生之後，必須由連結程式(Cross-Linker 13)將所有目標程式聚集起來，加以連結，使許多小程序式模組能結合成為一個完整的絕對目標程式(Absolute Object 14)，以便執行運作。目標程式00除了可以直接參加連結之外，還可以經過目標程式庫管理程式(Object Librarian 11)的運作而被置入目標程式庫(Object Library 12)中。此一目標程式庫可在程式連結時，提供庫中已預存的目標程式予

連結程式，因此亦為儲存軟體元件的一個處所。

6. 程式模擬

絕對目標程式04可以不必在實際的微電腦機器上運作，而可經由模擬程式(Simulator 15)的協助，直接在小型以上電腦加以模擬，而達到偵錯除錯的目的。這種作法的好處在於不必將程式傳送至微電腦發展系統之中，就可以進行測試作業，不但節省測試時間，而且可以減少微電腦發展系統的工作負荷，同時也可以即時在線上加以修改，省却麻煩的修改程序。模擬程式是一個極為有力的測試工具，尤其對於軟體測試可說具有百分之百的可靠性，值得加以推廣使用。

7. 程式移轉

雖然模擬程式中具備有相當程度的輸入輸出(Input / Output)及中斷(Interrupt)之模擬能力，可是一旦所測試的程式須作即時測試(real-time test)或是和實際的硬體線路相連接測試時，則模擬程式就無法負擔此項工作，這時就必須將絕對目標程式經由通訊程式(Communication Package 16)的協助，將之傳送到微電腦發展系統之中來進行此類測試。通訊程式可說是小型以上電腦系統以及微電腦系統之間的橋樑，二個系統之間可以藉著此一橋互相傳送資料，因此就可將二個獨立的系統連結在一起。

8. 程式測試

當絕對目標程式04被傳送至微電腦系統中後，可利用線上模擬器(In-Circuit Emulator 17)來進行即時測試或整體測試，在做此種測試時可說是對微電腦系統資源最有效的運用，因此不致於產生資源浪費的現象。當程式完全測試完成，沒有發現錯誤之後，可依實際作業上的需要，經過僅讀記憶體記錄器(Prom Programmer 18)之作用，將程式記錄於僅讀記憶體中，完成整個程式發展程序。

以上所提的是一個微電腦程式在整體發展系統中的一個典型發展程序，其程序和傳統式微電腦發展系統差不了多少，惟一不同的就是大部份很花費

時間，而且浪費資源的工作均被移到小型以上電腦中進行，大幅度提高程式發展速度，提高生產能力。

在下一章中，我們對各個工具程式所須具備的能力，提出一個基本的建議，以做為選取、評估，或設計製造這些工具的一個參考。

四、發展工具 (Development Tools)

在本章中，我們將探討上一章所提出的各種軟體工具所須具備的基本能力，這些基本能力有些並非絕對必需，但是若要做為一項有效的工具，則應具備有大部份本文所提出的能力不可，否則其功能必然會不足。

1. 編輯程式(Editor)

編輯程式的基本目標在於協助操作人員將文件或程式鍵入電腦之中，由於打字是一項費時而且較無創造性的工作，因此編輯程式的能力就格外重要。以程式發展的經驗來看，鍵入文件及程式往往佔據了很多的時間，最大的忌諱就是由程式人員或工程師自己鍵入大量的文件或程式，因為這些人員並不具有打字的專長，因此不但花費極長的時間打字，同時往往錯誤極多，延緩了程式發展的速度。

編輯程式常見的有二種，一種是較老式以行為單位 (Line Editor)，另一種則以整頁螢幕為作業單位 (Page Editor)，在實際的作業上，以整頁為單位的編輯程式其效能要比以行為單位的編輯程式好得多，因此我們建議能多利用以頁為單位的編輯程式。

編輯程式所須具備的基本功能大約如下：

- (1) 標點控制(Cursor Control)：包括標點之上、下、左、右移動、跳字、跳句、跳行，移到起點或終點、跳段、跳頁等，主要是移動標點作為指示之標的。
- (2) 字句尋找(Search)：包括設訂所要找尋的目標，設訂所要找尋的範圍，以及執行正式的找尋

作業。

- (3) 修改字句(Modification)：修改字句的範圍極為廣泛，包括加入、去除、替換等，而修改的目標又可為字、句、行等，這個部份其實才真正是編輯作業的執行者。
- (4) 段落移改(Block Operation)：在實際的作業中，往往須要進行對於整段資料的刪改或移位，因此這種功能也是不可或缺的，否則對於段落的更改就極為費時費事了。
- (5) 集指令功能(Macro)：有些固定但卻冗長的指令群可集合而成集指令，如此就可以減少大量不必要的按鍵次數，同時也可減少錯誤。
- (6) 資料儲存(Data Storage)：在編輯資料之時，往往須要將部份資料存入另一磁碟檔案之中，因此編輯程式就必須具備此一功能，否則極為不便。

以上所提為主要的項目，另外還有一些文件編輯功能、終端機規格設定功能、反複功能、協助功能等都能視需要加入，以協助使用者能有效運用此一工具。編輯程式的設計必須以易懂、易學、易用為原則，務必使此一工具顯得和善可親，則必能消除輸入瓶頸。

2. 文件印製程式(Formatter)

文件印製程式的功能在於印出整齊美觀的書面文件資料，亦即一般所謂的文件處理工具 (Word Processing Tool)。整齊的文件對於溝通及調節心情有極大的關聯，同時對於查索及文件運用也有助益，因此不可輕易加以忽視。

文件印製程式至少須具備下列四種功能方可：

- (1) 資料整理(Text Formatting)：這個功能主要是設定文件內容的印製方式，例如：隔行、空行、跳頁、成段、留下圖面空間、加入註腳、將題目置於中間位置、字體放大縮小、搜集專有名詞並加以分類等皆為實用而不可缺乏之功能。

(2)格式設定(Page Formatting) : 這個功能主要是用來安排每頁文件的格式。例如: 加印文件之名稱、編列頁數、印製日期等, 這些項目都是印製於每頁文件的上端或下端, 以資作為辨識之用。

(3)型式控制(Mode Setting) : 這個功能主要用以控制文件內容各行各段的處理方式。例如: 每行右邊必須對齊, 大小寫字體的控制, 每句後所必須留置的空間, 以及每一段前所須預留的空間等。

(4)參數設定(Parameter Setting) : 此一功能最主要用以控制印製文件的外觀。例如: 每頁文件所應印製的行數, 每一行所應印製的字數等。

在以往, 資料文件的處理往往為一般人所忽視, 近年來已有極大改善, 文件印製程式僅只為諸多手段中的一種而已, 這方面的技術近年來突飛猛進, 是一具有強大潛力的市場, 今後我們實在不應該加以忽視。

3. 高階語言編譯程式(Cross-Compiler)

高階語言編譯程式的功能就在於將高階語言原始程式轉譯為組合語言程式, 關於高階語言編譯程式由功能較為固定, 較多的變化僅在於錯誤訊息的顯示以及印製報表方面而已, 而且一般人都已有相當的經驗, 因此在這裏並不多加說明, 讀者可自行使用一兩種編譯程式, 自能有所體會。

4. 組合語言編譯程式(Cross-Assembler)

組合語言編譯程式的目的在於將組合語言原始程式轉譯為目標程式, 由於組合語言的指令內容極為廣泛, 而且較不易使用, 因此編譯程式往往提供許多功能, 以協助程式設計人員克服困難, 設計出正確的組合程式來。

組合語言編譯程式須具有下列功能:

(1)資料定義(Data Definition) : 由於最初的組合語言中沒有資料類別(Data Types)的觀念, 因此使用起來非常不方便, 而且對於資料方面的檢查及管制也非常鬆弛, 因此在編譯程式中乃

提供資料定義的功能, 以彌補此一缺失, 常見的資料類別有: 位(Byte)、字(Word)、整數(Integer)、點浮數(Floating-Point Numbers)、十進位數字(Decimal Numbers)等, 值得一提的是現在十六微電腦中已開始提供基本的資料類別了。

(2)記憶體預留(Memory Reservation) : 這類功能最主要的功用是預留資料儲存空間。

(3)標名定義(Symbol Definition) : 這個功能的目的是在於將標名與一些具體的事物(例如記憶體位置或實際數值)相聯結, 以利使用。

(4)條件式轉譯(Conditional Assembly) : 此一功能的目的在於設定及測試各種轉譯條件, 以決定到底應該轉譯組合程式的那一部份, 最常見的結構就是 IF-THEN-ELSE 句型。

(5)列印控制(Listing Control) : 此一部份功能最主要用於控制轉譯後列印之報表格式, 錯誤訊息, 對應表(Cross Reference)等, 同時還必須能有跳頁、標題(Title)等控制功能, 能使報表更適合閱讀。

(6)程式段落(Program Segmentation) : 此功能之目的在於控制程式模組的類別, 這些模組大致上分為絕對模組(Absolute Module)及未定位模組(Relocatable Module), 其中各類模組又可分為程式段落(Program Segment)以及資料段落(Data Segment)兩類。

(7)程式連結(Program Linkage) : 此一功能提供模組相互連結的能力, 包括模組命名, 共用資料(Public Data)及外在資料(External Data)之定義, 這個功能在模組化程式製作(Modular Programming)的作業方式中極為重要, 如果沒有這個功能, 大部份程式均將成為龐然巨物了。

(8)集指令(Macros) : 這個功能其實可分為二個部份, 一部份是用以指導編譯程式如何去集指令

資料庫中選取所指定的集指令。另外一部份就是用以讓程式製作人員在程式中自行定義程式中所用的集指令。

由於組合語言本身能力有限，因此編譯程式就變得極為複雜，這是無可避免的事，因此就增加了程式設計人員的負擔，如果不是非常必要，應儘可能避免使用組合語言，如此不但程式邏輯容易看清楚，而且可以大幅度縮減程式發展及維護費用，是較為明智的作法。

5. 連結程式(Cross-Linker)

連結程式的目的在於將已編譯好的目標程式模組連結成爲一個完整而可執行的絕對目標程式，只有經過這個程序才能將公用標名(Public Symbols)以及外界標名(External Symbols)相互對照結合，而不致發生有名無實的現象。除了絕對目標程式之外，連結程式還能夠產生對應表(Load Map)，以供程式偵錯及測試時之用。

連結程式的外在功能極為簡單，與使用者之間也沒有很複雜的介面須要討論，唯一與小型以上電腦不同之處在於需要提供僅讀記憶體(ROM)以及讀寫記憶體(RAM)之起始位置，如此可給予微電腦設計人員較大的方便，並減少錯誤，這個功能如能與模擬程式或線上模擬器配合起來，當更爲有效。

6. 目標程式庫管理程式(Librarian)

此一發展工具的作用在於管理目標程式庫，其主要的功能大致上應包含加入、取出、修正、取代各種目標程式模組，以使目標程式庫隨時都保持最正確而且最有效的狀況。

目標程式庫是一個極為有效的軟體元件組裝工具，但是往往受到忽略，以致使生產力量重複使用在相類似的工作上，形成一種極大的浪費。軟體元件的觀念在於建立有系統、有組織，而且功能完備的軟體程式群，這些元件經過長時間的發展使用，必然能夠具備高度的能力及可靠性，如果能將這些元件有計畫的納入程式庫之中，不斷積聚，必然能

够形成一強而有力且功能完備的資料體系，不但使以往所累積的經驗及工作成果不致失散，而且可替未來的各項計畫節省大量人力及時間，只要能持之以恆，長時間不憚其煩的去做，必然能收到宏大的效果。

7. 模擬程式(Simulator)

模擬程式的功能就是在小型以上電腦中，模擬微電腦之動作，以達到測試及偵錯之目的。由於微電腦的功能日趨於複雜，所以模擬程式也日漸繁複，所幸偵錯所須指令內容大體上仍爲一定，所以還不會造成使用者太多的困擾。

模擬程式所須具備的功能大致說明如下：

- (1) 環境設定(Enviroment Defiltion) : 用於設定所要模擬的作業環境，最主要的就是僅讀記憶體及讀寫記憶體的範圍，輸入輸出通道的範圍，各種輔助元件之提供與否等項目，這個功能由於硬體元件的日新月異，已有不能負荷之趨勢，例如數學運算元件(Arithmetic Units)，記憶體管理元件(Memory Management Units)等都是很明顯的例子，尤其以十六位元微電腦爲甚。
- (2) 程式執行(Program Execution) : 包括執行之起始點控制，單一指令執行控制，停頓(Break)之控制，追蹤(Trace)之控制，其中停頓及追蹤控制又包含對記憶體，輸入輸出通道，暫存器等讀寫狀況控制，這個功能對於程式偵錯非常重要。
- (3) 資料處理(Data Manipulation) : 這個功能的目的是在於處理記憶體及暫存器內的資料內容，包括顯示及更改資料，同時還能將某些指定的資料存入磁碟檔案之中，以便日後查詢。資料的輸出輸入方式應該能由使用者加以控制，依照資料內容及使用者的需求，以指令、二進位、八進位、十進位及十六進位等方式加以處理。
- (4) 標名處理(Symbol Manipulation) : 使用者應該能根據自己的需要而設定各種標名，同時賦予數值，直到不須要時再予取消，在偵錯時能發

擇很大的效用。

(5)集指令處理(Macros)：有些模擬操作程序可以事先設定成爲集指令預存起來，這樣在操作時可以省却許多麻煩，同時這些集指令可以和某些特定的動作連結起來，只要符合這些事先指定的狀況時，模擬程式就自動執行某一些特定的集指令。這些狀況包含對記憶體，輸入輸出通道，暫存器的讀寫，或是模擬程式通過某一停頓點或追蹤點，這一功能能提供使用者強而有力的工具來追蹤及記錄各種所發生的狀況，是一極爲重要的功能。同時在集指令之中也可以加入條件執行的能力，也就是具有 IF-THEN-ELSE 的條件句，如此可以使集指令顯得更有彈性，能適應各種使用者所預期的狀況，發揮最大的功能。

(6)輸入輸出(I/O Operations)：在許多狀況下，程式須和外界環境有所交通，因此模擬程式必須具備與外界資料進出的功能。這一功能必須具備有建立通道，關閉通道，讀取及送出資料的功能。通道必須能和所有可能的週邊裝置連結起來，尤其是磁碟，報表機以及終端機最爲重要，對於磁碟檔案之處理方式在基本上必須具有順序存取(Sequential Access)的功能，如果能具有隨機存取(Random Access)的功能就更好了。由此一功能受限於模擬程式所處的作業系統環境，因此往往也具備了此一作業系統中輸入輸出系統的模式，所幸小型以上電腦內的作業系統均極有力，因此所提供的功能大體上均能滿足使用者的要求。

(7)中斷處理(Interrupts)：中斷處理在模擬程式中是最難以克服的一個大難題，由於中斷是外界所送入之訊號，同時又是一種非同步訊號，因此極難模擬。目前最常見的作法是提供一組暫存器作爲中斷處理機構，其中包含了各種輸入的中斷訊號，各種障蔽(Mask)，以及向量位置(Vector)等，同時提供一組指令來設定各種中斷訊號。在模擬程式模擬各個指令之前，必先檢查各個中斷訊號，若其中有任一訊號已被設定，

則開始執行此一訊號所對應的服務程式(Interrupt Service Routine)，待執行完畢之後才又重新回復被中斷前的狀態，繼續模擬各個指令。由於微電腦的中斷處理也日益複雜，因此有些模擬程式也提供模擬中斷控制元件(Interrupt Controller)的動作，以符合實際的狀況，這也是模擬程式所須克服的一項問題。

由於模擬程式並非真正處理實際的硬體組件，因此碰到與硬體組件有關的事項均未能有令人滿意的結果，尤其在組合測試(Integration Test)時，都不能達到目的，因此在整個發展環境中仍須有線上模擬器的存在，以補模擬程式的不足。不過若僅只發展軟體程式，而與硬體組件沒有太大關聯時，則模擬程式仍爲一極爲有效的發展工具，可以提供軟體發展人員偵錯及測試的程序，因此模擬程式的存在仍然有其必要。

8.通訊程式(Communication Program)

通訊程式是使用於微電腦發展系統之中，其目的在於使小型以上電腦與微電腦系統之間能夠互相交換資料，以便將程式置於所欲使用的發展環境之中。資料的交換以最簡單的方式爲之，只要能將磁碟檔案中的資料傳送對方的磁碟檔案內即可。

爲了使用及操作方便起見，通訊程式通常包含下列三種狀態：

(1)設定狀態(Setup Mode)：處於這種狀態之中即可設定二個系統之間的各種通訊特性。例如：傳送速率，編碼方式，設訂暫停及回復碼，設訂停止傳送碼，設訂資料編輯方式等，有了這一狀態則微電腦發展系統將可連接各種小型以上電腦發展系統，增加此一通訊程式的彈性。

(2)終端機狀態(Terminal Mode)：處於這種狀態中，則微電腦發展系統就如同小型以上電腦的一個終端機，操作人員可以在微電腦的終端機前輸入小型以上電腦的各種作業指令，而不須使用真正連接於小型以上電腦的終端機來作業，在實際的作業中，這個狀態的存在可提供使用者極大的便利。