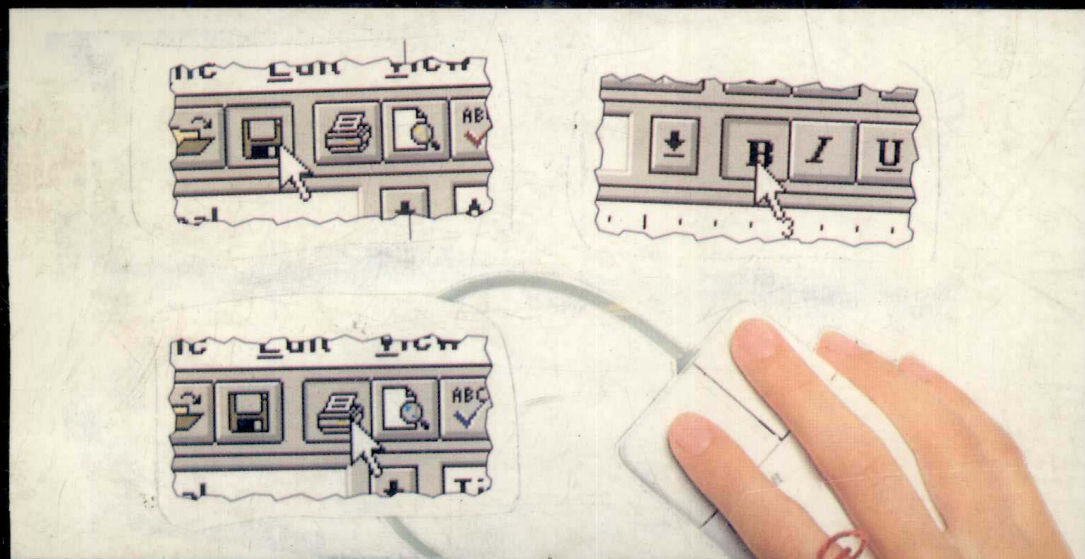


计算机语言技术系列丛书(二)

# WINDOWS

## 程序设计实务(一) 基础篇

李元泰 著  
施威铭



旗標  
FLAG

学苑出版社

希望

计算机语言技术系列丛书(二)

# WINDOWS 程序设计实务(一)

## 基础篇

李元泰      著  
施威铭  
万 博      改编

学苑出版社

**(京)新登字 151 号**

**内 容 简 介**

本书循序渐进地介绍用 C 设计 Windows 应用程序的技术,介绍了 Windows 程序的组织结构、窗口的基本输出、窗口的键盘输入、窗口的鼠标输入、窗口的计时器输入、窗口资源、图形资源、菜单资源、对话框、GDI 入门及字型与打印。本书适用于使用各种 C 编写 Windows 应用程序的读者使用。

欲购本书的用户可直接与北京海淀 8721 信箱书刊部联系,电话 2562329,邮政编码 100080。

**版 权 声 明**

本书由旗标出版有限公司授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

计算机语言技术系列丛书(二)

**WINDOWS 程序设计实务(一)**

**基础篇**

---

著 者:李元泰 施威铭  
改 编:万 博  
责任编辑:陆卫民  
排 版:万博图书创作社  
出版发行:学苑出版社 邮政编码:100036  
社 址:北京市海淀区万寿路西街 11 号  
印 刷:双青印刷厂  
开 本:787×1092 1/16  
印 张:20.5 字数:475 千字  
印 数:1~5000 册  
版 次:1994 年 9 月北京第 1 版第 1 次  
ISBN7-5077-0905-1/TP·29  
本册定价:38.00 元

---

学苑版图书印、装错误可随时退换

## 原 序

跨入计算机写作的领域,对我而言,实在是一个命运中的巧合,以前甚至连作梦也想不到的。

一个身处异乡的留学生,却念念不忘国内的计算机发展,说来实在有点多管闲事了。本来嘛!使用计算机只是我的一个业余爱好,虽然从APPLE时代至今,玩了十年了但从未动过以此谋生或出书的念头。在美国开始学习Windows编程,实在是有感于这个操作系统在美国太盛行了,以我目前就读的加州理工学院为例,学校的计算机中心共有PC、Macintosh、Sun SPARC、Next四个开放实验室,PC实验室中所有的486,都被设置为开机后立刻进入Windows,各办公室的独立PC也莫不如此。翻开计算机杂志,琳琅满目地全是Windows相关产品,除了Game之外,DOS程序简直已经没有立足之地了,计算机世界似乎早已被一个个窗口所征服了。不学学Windows编程,似乎就难以在未来的世界中生存。

去年八月底回台湾一趟,顺便逛逛书店,意外发现书中的Windows编程的书少得可怜。好不容易找到两本带回美国,却发现其中文字脱漏、语句不通、错误比比皆是,心想:难道台湾的计算机产业就这么可怕,非要读这种东西不可吗?于是就自不量力,想以自己的学习经验,写一本入门书,如果对目前正在门外徘徊的人有点帮助,也算是积点功德了(当然,赚点稿费,贴补一下学费,也是好的)。

真正动笔后才发现其中的难处:Windows包含的内容、技巧太多了,想要兼顾简易性和完整性,几乎是不可能的事。数度想要放弃,最后还是在背负着课业压力下完成了,回想那几个月的心力交瘁,到现在还有点不敢相信,自己居然能走过这段路。

感谢旗标施威铭先生不嫌弃我这后生小子,以及旗标出版有限公司及研究室的先生小姐们全力的支持,完成校对测试设计等工作,使这本书终于能够出世,在此对于他们的认真负责与敬业精神,致以十二万分的敬意与谢忱。真的,完成一件事,才发现其中受惠实在太多。我想,大家也都是抱着推广资讯普及、帮助学习计算机人的理想,在从事这份工作,许多话也无须多说,只要往后更加努力,推出更好的作品,也就是一种回报了。

一个人在异地读书、写作,感觉上当然是非常寂寞了。感谢胜慧从不间断的信件、电话的安慰、鼓励,以及无悔的支持,才使我有勇气度过无数挑灯夜战的孤独夜晚,忍受一切辛劳,如果本书能有任何一丁点的成就,我都希望能与他一起分享。

愿将所有荣耀与掌声,献给父亲,以及在天上的母亲。至于批评及建议,则是作者自己希望能够获得的。

李元泰

1993.4.9

于美国加州理工学院

## 施 序

当 Windows 操作系统广被用户喜爱的同时,我们也常听到“Windows 编程不易学习、障碍重重、难以克服……”等种种说法。

Windows 的编程和以往传统的方式是有些不同,它隐含了面向对象(Object Oriented)、事件驱动(Event Driven)和信息传送(Message Passing)等观念,因此对于习惯于传统编程思考的人而言,是必须花时间来适应及学习的。除了观念的革新之外,Windows 系统本身复杂而强大的功能,也常令学者花更多的时间来熟练与掌握。学习 Windows 编程的障碍主要就是以上原因所致。

为了提供 Windows 编程一条正确而有效的学习路径,旗标出版公司特别规划了一套 Windows 编程实务丛书。其内容包含:

- ☐ Windows 编程实务(一)基础篇
- ☐ Windows 编程实务(二)进阶篇
- ☐ Windows 编程实务(三)IPC 篇
- ☐ Windows 未公开的秘密

我们希望这套书能提供学习者一个顺畅、高效率的学习课程,不用花费无谓的精神与时间即能掌握 Windows 编程的原理与精华。

李元泰先生和我编写了基础篇的部分,但我因事务繁忙,除了动笔写作第 1、2、7 章之外,其它章节多为李兄的功劳,而我和研究室同仁只不过做些测试、修饰的工作而已。本书从交稿到出版,其间为了测试、查错、内容的安排、文章的修改、不断的沟通与讨论,足足花了一年的时间,更占用元泰兄每次回国短暂的时间,在此特别对他的耐心致上我们的歉意!但我们都有同样一个目标,那就是要把一份好的作品呈现给读者!我想,就是因为这样的理想,才能使我们在这一年内持续努力而完成本书吧!

为了使本书的内容更臻完善,我们需要您宝贵的意见,任何看法都是我们需要的!

施威铭

1993/11/12

# 目 录

|                                                       |    |
|-------------------------------------------------------|----|
| 原序 .....                                              | I  |
| 施序 .....                                              | I  |
| 热身运动.....                                             | 1  |
| <b>第一章 第一个 Windows 程序:传说</b> .....                    | 5  |
| 1.1 最小的 Windows 程序 .....                              | 6  |
| 1.2 Instance 与 Handle .....                           | 7  |
| 1.3 WinMain()和它的 4 个参数 .....                          | 9  |
| 1.4 第二个 Windows 程序:再论 hInstance 与 hPrevInstance ..... | 12 |
| 1.5 显示标准的窗口 .....                                     | 15 |
| <b>第二章 Windows 程序的组织结构</b> .....                      | 19 |
| 2.1 单工与多工作操作系统.....                                   | 20 |
| 2.2 消息 Message .....                                  | 22 |
| 2.3 窗口函数:消息所要送达的对象 .....                              | 24 |
| 2.4 Windows 程序的流程 .....                               | 28 |
| 2.5 程序分析(一):WinMain() .....                           | 31 |
| 2.6 程序分析(二):窗口函数 WndProc .....                        | 32 |
| 2.7 模块定义文件.....                                       | 34 |
| 2.8 编译运行.....                                         | 35 |
| 2.9 Hunfarian 命名规则 .....                              | 38 |
| <b>第三章 窗口的基本输出</b> .....                              | 41 |
| 3.1 Windows 的窗口工作原理 .....                             | 42 |
| 3.2 工作区与绘图结构.....                                     | 42 |
| 3.3 GDI 绘图与范例.....                                    | 44 |
| 3.4 文本的输出.....                                        | 52 |
| 3.5 滚动条.....                                          | 58 |
| <b>第四章 窗口基本输入(一):键盘</b> .....                         | 65 |
| 4.1 DOS 与 Windows 不同的地方 .....                         | 66 |
| 4.2 键盘消息.....                                         | 66 |
| 4.3 按键光标:Caret .....                                  | 71 |
| 4.4 范例:EDIT DEMO .....                                | 72 |
| <b>第五章 窗口基本输入(二):鼠标</b> .....                         | 79 |
| 5.1 鼠标的基本要素.....                                      | 80 |
| 5.2 另一个例子 MCLICK .....                                | 84 |
| 5.3 捕捉工作区以外的鼠标消息.....                                 | 88 |

|             |                                  |            |
|-------------|----------------------------------|------------|
| 5.4         | 鼠标的形状.....                       | 92         |
| <b>第六章</b>  | <b>窗口基本输入(三):计时器.....</b>        | <b>97</b>  |
| 6.1         | 窗口的计时器.....                      | 98         |
| 6.2         | 如何使用计时器.....                     | 99         |
| 6.3         | 范例:窗口的监视器.....                   | 105        |
| <b>第七章</b>  | <b>窗口资源简介.....</b>               | <b>113</b> |
| 7.1         | 何谓资源 .....                       | 114        |
| 7.2         | 资源的种类 .....                      | 115        |
| 7.3         | 资源开发工程介绍:Resource Workshop ..... | 117        |
| 7.4         | 资源应用范例 .....                     | 126        |
| 7.5         | 资源描述语言 .....                     | 127        |
| <b>第八章</b>  | <b>图形资源.....</b>                 | <b>131</b> |
| 8.1         | 位图资源 .....                       | 132        |
| 8.2         | 使用图形资源 .....                     | 133        |
| 8.3         | 画出位图 .....                       | 138        |
| <b>第九章</b>  | <b>菜单资源.....</b>                 | <b>147</b> |
| 9.1         | 菜单描述文件的语法 .....                  | 148        |
| 9.2         | 在窗口中加上一个菜单 .....                 | 150        |
| 9.3         | 在程序中建立菜单 .....                   | 153        |
| 9.4         | 菜单消息处理 .....                     | 156        |
| 9.5         | 菜单函数 .....                       | 157        |
| 9.6         | 系统菜单及弹出式菜单 .....                 | 165        |
| 9.7         | 加速器资源 .....                      | 172        |
| <b>第十章</b>  | <b>对话框.....</b>                  | <b>175</b> |
| 10.1        | Windows 内建的控件 .....              | 176        |
| 10.2        | 按钮;BUTTON .....                  | 182        |
| 10.3        | 文本编辑框;EDIT .....                 | 184        |
| 10.4        | 列表框.....                         | 187        |
| 10.5        | 滚动条;ScrollBar .....              | 189        |
| 10.6        | 静态字符串;Static String .....        | 190        |
| 10.7        | 对话框资源.....                       | 190        |
| 10.8        | 对话框消息及处理函数.....                  | 193        |
| 10.9        | 对话框的使用和种类.....                   | 195        |
| <b>第十一章</b> | <b>GDI 入门 .....</b>              | <b>211</b> |
| 11.1        | 什么是 DC (Device Context) .....    | 212        |
| 11.2        | 绘图对映模式.....                      | 216        |
| 11.3        | 窗口的颜色.....                       | 225        |
| 11.4        | 绘制图形.....                        | 232        |
| 11.5        | 位映射运算简介.....                     | 248        |

|                                                    |     |
|----------------------------------------------------|-----|
| <b>第十二章 字型与打印</b> .....                            | 253 |
| 12.1 文本输出.....                                     | 254 |
| 12.2 字型种类与使用.....                                  | 256 |
| 12.3 打印机 DC .....                                  | 272 |
| 12.4 打印输出.....                                     | 277 |
| <b>附录 A Borland C++ 3.1 的目录与文件结构</b> .....         | 289 |
| <b>附录 B Microsoft C 7.0/SDK 的目录与文件结构</b> .....     | 291 |
| <b>附录 C 如何使用本书范例</b> .....                         | 295 |
| <b>附录 D 本书各章使用的 WINDOWS 函数</b> .....               | 297 |
| <b>附录 E MessageBox 与 WinExec 函数</b> .....          | 309 |
| <b>附录 F WINDOWSX.H 与 message crackers 简介</b> ..... | 313 |



# 热身运动

## 欢迎进入 Windows 编程的领域

欢迎各位,一起来进入 Windows 编程的领域

计算机硬件的突飞猛进,PC 的 CPU 自 8088、286、386、486、486DX— I、甚至最新的 Pentium,功能之强大,已可追述到早期的主机系统。不但让我们拥有速度愈来愈快、品质愈来愈高、功能愈来愈强的机器,也带动了“多任务操作系统”及“圆形作用环境”的发展与流行。Windows 便是这种潮流下的产物。

虽然设计 Windows 程序,并不是一件很简单的事,但也绝不会像传说中那么困难。只要将基本概念弄清楚,熟悉“事件驱动”式的思考方式,想学会设计 Windows 程序,并不是太难的事。更何况,Windows 提供了大量强有力的工具和函数,灵活运用这些函数,反而可以节省许多细节上的工作。Windows 同时还有许多特点,本书会一一指出,并引导读者逐步的进入 Windows 的世界。

本书虽是 Windows 编程的基础引导,但并不是一般编程的入门书,如果您连最基本的编程都不会,那么最好能够先将基本的 C 语言弄清楚了,再来阅读本书。

## Windows 编程的背景知识

学习 Windows 编程,有一些基础的知识和工具是不能缺少的。在开始进入正式的内容之前,我们将一些背景知识和开发工程先做一个简介。倘若您未准备齐这些,请赶紧去补足,如此才能顺利地学习下去。

### 什么是 Windows

什么是 Windows? 如果您连 Windows 都还没见过的话,建议您先暂时丢开本书,买一套原版的英文或中文 Windows 用用。目前在美国的零售市场,有低到美金 79 元以下的折扣价(不到台币 2000 元),真可谓物美价廉(听说台北的光华商场也并不)。单是其中附带的十几个应用程序,如 WRITE、PAINTBRUSH、CARD、SOLITARI,便已物超所值了!

学习 Windows 编程,将会使您在计算机世界中,迈入一个崭新的领域。

### 要先熟悉 Windows 的操作

最重要的是,如果连 Windows 的操作经验都没有,那很显然本书现在并不适合您。您不但要有 Windows 系统,更要能用上一阵子,熟悉 Windows 的一些操作特性和反应。图形界面环境(不光是 Windows,Macintosh、NeXT、X—Windows 等都一样)非常重视信息及事件的处理。所以在动手设计程序之前,您必须对一些操作的反应过程充分了解才行。而最好的方法是使用鼠标或键盘,随兴地玩一阵子,试试在窗口的任何一个部位按钮、拖动(Drag)、或双击(Double-click),看看有些什么样的反应,甚至将这些动作的细节抓出分解。当然做不到这一步也没有关系,最主要是心中要有个概略的轮廓。

## 要学会 C 语言

Windows 的主要开发环境是 C 语言,大部分的 Windows 程序代码是用 C 写成的。因此,您要阅读本书,必须有 C 语言的基础。C 是一种很适合发展的操作系统的“中阶”语言,速度也不错,但此处建议您必须会 C 语言,并不代表 C 比其他语言更优秀,笔者并非“C 至高无上论者”,我们适用 C 语言,乃是因为 C 是 Windows 的母语,唯有使用 C,方能用得上 Windows 所提供的所有函数,发挥 Windows 编程的全部威力。

如果您学过其他语言,但不会 C 语言,建议您不妨先找本 C 的教程来读一读,施威铭先生的《C 语言实务》是个不错的选择,其他市面上的书也很多,多学一种语言有益无害。

学习 C 并不是件困难的事,只要您知道 C 的基本用法,流程控制及指针和结构的概念,阅读本书便不成问题。

## 安装一个开发环境

Windows 编程比起其他 DOS 程序的发展,最大的障碍,便是它需要一个复杂且较昂贵的发展环境。

随着每一版本 Windows 的推出,Microsoft 都会提供更新的 SDK (Software Development Kit),以供设计者开发应用程序,SDK 要配合微软自己的 Microsoft C 编译器,并且至少要在 5.1 版本之上,其他早期版本就不行了。

使用 Microsoft C 配合 SDK 开发程序,老实说并不是很愉快的事,即使您的系统已正确安装(要设置一大堆环境参数),每写一个程序,其辅助文件需要十来个也不足为奇,编译手续实在繁杂。

但不可讳言,Microsoft C 和 SDK 虽然麻烦而费时,但其高度可变性、最优化能力和 Windows 程序代码的编译能力,多年来一直是 Windows 程序员唯一且可靠的选择。

至于另一 C 语言的霸主—Borland International 公司,一直以其快速方便的 TURBO C 称霸市场,目前该公司有两个系列的产品:BC++与 TC++。

## 开发 Windows 程序的利器 BC++3.0/3.1

Borland C++ 3.0/3.1 版的推出,可说真正是向前迈进了一大步。以 Borland C++3.1 版为例,它的 BC for Windows (简称 BCW)不但可以直接在 Windows 上编写、编译、运行 Windows 程序,且拥有最优化功能及发展 Windows 3.1 版程序代码的能力。对初学者来说,笔者认为 BCW 3.1 版还有下列好处:

- 一、对于简单的程序,毋需撰写模块定义文件,不须设工程管理(project)文件或.MAK 文件,只要直接运行 RUN 功能,系统会自动补足您未设置的文件,并马上看到结果,就和使用 TURBO C 时一样方便。
- 二、BCW 具有一项关键字着色的新功能,也就是对程序的关键字、保留字、数据等,会自

动加上颜色或粗细字体。这样可以马上一目了然,减少许多输入上的错误;调试会更加方便,可以马上看出整个结构,减少 Trace 的时间。

三、BCW 的套件中,另有许多工具。对 Windows 编程很有帮助,包括 TDW(调试工具)、Resource Workshop(资源工作室简称 RW)及可观察系统信息流向的 WinSight 等等,都是极佳的辅助工具。

四、不需要另备 SDK。

笔者并非 Borland 公司的拥护者,但 Borland C++ 3.0/3.1 版对初学者来说,比 Microsoft C+ SDK 好用得多,乃是不可争辩的事情。本书所有范例,均在 Borland C++ 3.1 下发展及测试。当然,在 Microsoft C+ SDK 下亦可编译运行,但某些小地方(如不同函数库的使用)可能必须稍作修改。

### **适合大家使用的 TURBO C++ 3.0/3.1**

除了上述专业版本外,近来 Borland 公司为了争夺初学者市场,又推出了 TURBO C++ for Windows 3.X。这个编译器能直接在 Windows 上编译运行程序,但因为没有最优化的功能,所以价钱上也便宜得多。如果您初学 Windows 编程,又无力负担类似 Borland C++ 3.1 版之类超过三千元(美元)以上的高价级软件,建议您不妨购买 TURBO C++ for Windows,除了没有最优化功能外,其它在使用上几乎和专业版没多大差别。

另外还有一些其他的 Windows 开发工程,如 Visual Basic、Realizer 等。这些编译器虽不乏杰作,但总不如 C 语言一样能够发挥 Windows 所有的精华。其中 Visual Basic 由于易学、易用、有趣,已渐成一股风潮,但是如果您真想学会 Windows 编程,我们建议还是由 C 下手比较合适。

热身运动至此告一段落,准备开始了吗?

### **准备开始了...**

现在,我们真正要开始写一个 Windows 程序了!

Windows 编程,真的是一件复杂的工作,至少在刚入门的时候是如此,我们希望读者能不厌其烦地,把基本概念弄清楚,如此在真正进入编程的领域时,才能事半功倍,渐入佳境。

总之,事在人为,如果您已准备完成,就跟着我一起上路吧!

# 1

**第一个 Windows 程序：  
传说…**

传说:最小的 Windows 程序要 70 行以上,并且学好 Windows 编程,最快要 6 个月!

## 1.1 最小的 Windows 程序

现在,我们就来写一个最小的 Windows 程序吧!

```
#include <windows.h>
```

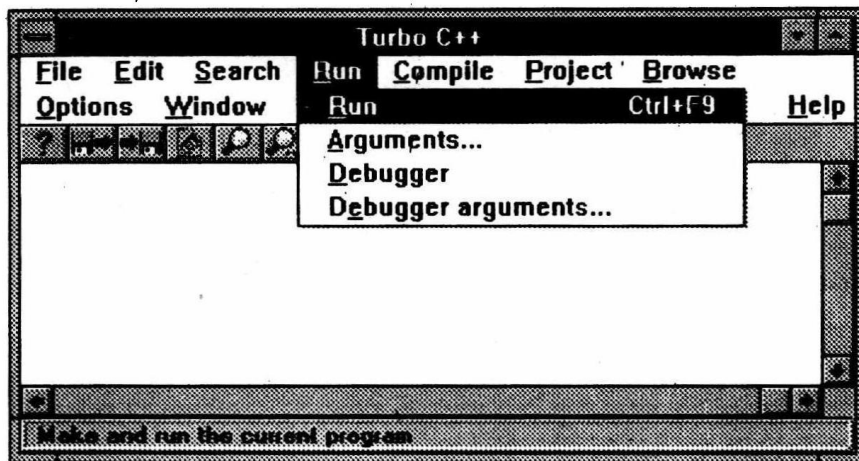
```
int PASCAL WinMain( HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR      LPCmdLine,
                   int         nCmdShow)
{
    WinExec("clock.exe", nCmdShow);
    return (NULL);
}
```

} WinMain() 的 4 个参数

程序 1.1 最小的 Windows 程序 MINI.C

这个程序除了开头的 `#include <windows.h>`, 其它就只有一个 `WinMain()` 函数, 并且 `WinMain()` 内部只有简单的两行语句。所以, Windows 程序也并不如传说中的困难!

如果你使用 TURBO C++ for Windows 或 Windows 模式下使用 BC++, 那只要选择 Run/Run 选项(或按 CTRL-F9 键), 系统就会编译、连接, 并将运行结果显示出来:



运行 Run/Run 之前要记得进入 Options/Directories 设置各工作目录

如果您使用 MSC, 则手续比较麻烦, 首先要参照 2.7 节的方式模块定义文件, 再依 2.8 节的方式来加以编译, 编译完成后再进入 Windows 用程序管理员的 RUN 命令来运行之。如果您是初学者, 我们建议勿轻易尝试 MSC, 因为学习 Windows 编程才是主要目标, 大可不必因操作而造成困扰, 不是吗?

**windows.h**

程序 1.1 一开始的 `#include` 命令包括了一个叫 `windows.h` 的标头文件(header file)。`windows.h` 是 windows SDK、TURBO C++ 或 Borland C++ 都会提供的一个标头文件,此标头文件定义了许多专供 Windows 使用的常量、函数原型、数据类型、数据结构等,以供编译 Windows 程序时使用。`windows.h` 的内容共有 5000 多行,内容丰富,有待您慢慢挖掘。最重要的是对任务 windows 程序而言, `#include <windows.h>` 这一行是不能免的。介绍完 `windows.h` 之后,接下去就是 Windows 程序的主体了,但是在进入主体之前,让我们先介绍两个很重要的概念:Instance 与 Handle。

**1.2 Instance 与 Handle****实例:Instance 的观念**

在 Windows 下,Instance 是一个有趣的概念。Windows 是一个多任务的操作系统,它不但可同时运行多个程序,即使是同一个程序,也可以同时重复运行。例如我们可以同时运行 3 个 write 程序编辑不同的文章,这 3 个运行中的 write 就是 3 个 Instance。在 Windows 下,同一个程序,可能同时就有几个 Instance 在运行。

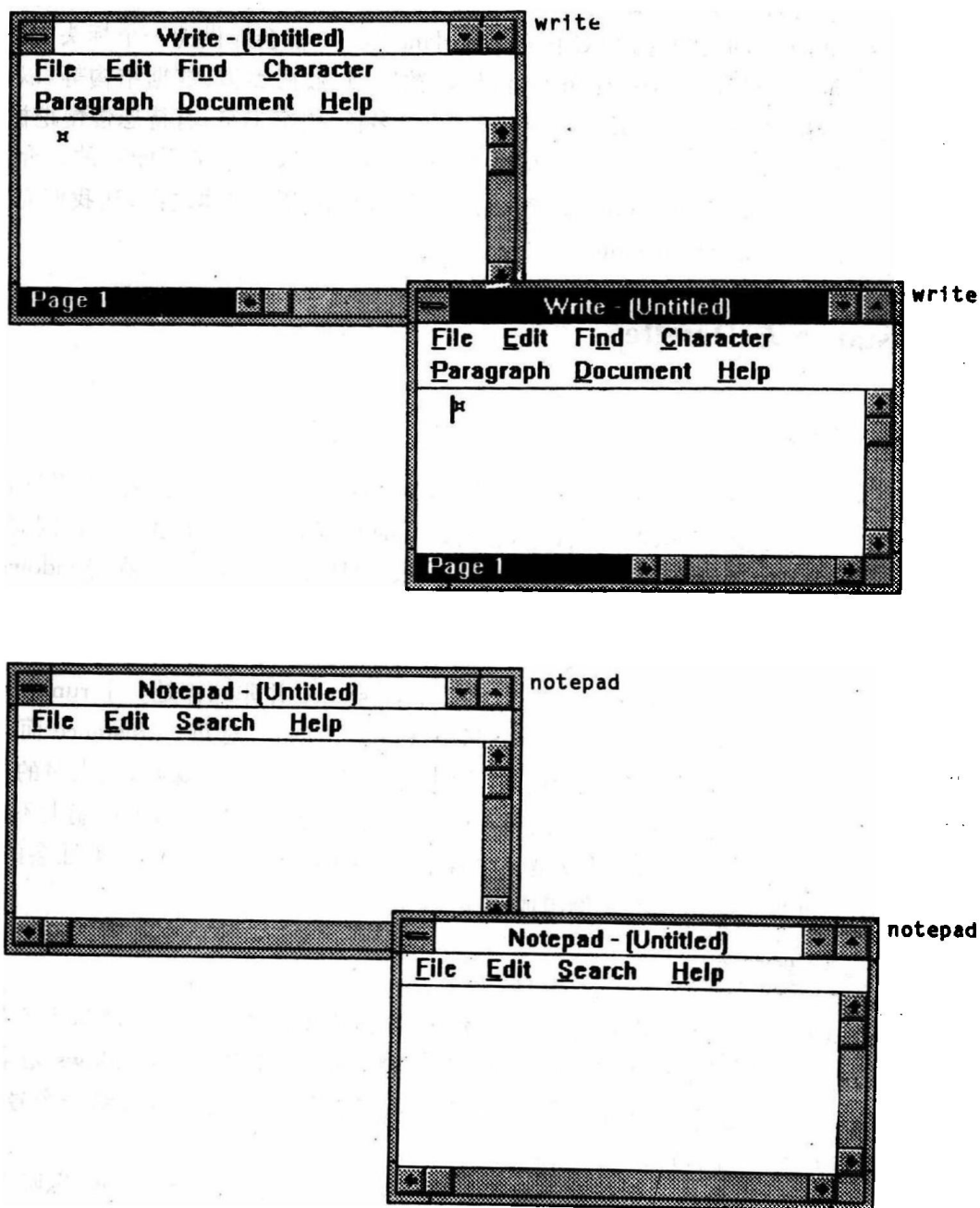
那么 Instance 应怎么称呼呢?有人称之为副本,有人称之为拷贝,因为 Instance 的英文解释为“... copy of a running program...”。但事实上,Instance 并非纯粹是一个 running program 的 copy。因为 Windows 的程序有两个部份:Code 与 Data,对每一 Instance 而言,其 Code 部分是相同,但 Data 部分则因运行的过程而各异,就好像兄弟或姊妹,初生时的外观、体格、环境均十分相似,但因人生旅途的机遇不同,结果在性格、思想、阅历也可能大不相同。Instance 便是这么一回事,Instance 可说是一个程序的“实例”,就好像组成人类社会的实例一般,Windows 的世界也是由这些实例组成而生生不息!

**句柄:HANDLE 的观念**

既然 Windows 允许一个程序同时运行多次,它就必须能够辨认运行中的每个实例(instance),但是 instance 是源自于同一程序的,所以无法用程序名来分辨。Windows 是采用挂号的方式来解决这个问题,它对每个运行实例(不管是否源自同一程序)都发给一个号码牌,这个号码牌称为 Handle(代码)。

事实上 Handle 的观念是一直存在于我们日常生活中的。例如到医院看病,医院会发给你一个号码;满 15 岁领身份证,户政单位会给一个身份证号码;开办公司时税务单位会给一个统一编号;买车时也要申请车牌号码...,这些都是 Handle。同样的,在 Windows 下,不但 Instance 有 Handle,窗口、设备、内存...也都有 Handle。

使用 Handle 有许多好处,例如身份证号码可区分同姓者,而医院的挂号牌更为重要,手术室的病人或产房的婴儿医生根本不认得,此时只有用挂号牌来验明身份了。并且若医院力量有限,使用号码牌可以随时了解目前患者人数,当 Handle 发不出来时,就是“满载”了,必须有人离去、交回号码牌后,才能再挂号。



2 个 write 和 2 个 notepad 的 instance

Windows 也是一样,每当运行一个程序,就会发给一张“挂号证(也就是 Handle)”凭着这张挂号证,才不会弄错要处理的对象。其它 Windows 系统中各类大大小小的资源对象:如窗口、菜单、对象框、对话框、按钮、笔、刷子、调色板、内存等,我们都必须先向 Windows 申请使用,如果申请成功,Windows 就会返回一个该对象的 Handle,藉由这个 Handle,我们才能



使用 Windows 系统的各种服务。往后您会发觉,Handle 在 Windows 大行其道,要调用 Windows 函数,没有 Handle 还真是寸步难行呢! 因为 Windows 系统的资源有限,这些使用的对象是 Windows 暂时“借”给我们的,使用完毕后,一定要记得释放,否则系统会因资源用完而无法顺利运转,“有借有还再借不难”是 Windows 编程初学者随时要注意的要诀。

Handle 是一个数(码),在 Windows 的程序中,用来存放 Handle 的参数,习惯上都以 h 为开头,如 hInstance 代表是 Instance 的 handle、hWnd 代表是窗口(Windows)的 handle、hMenu 代表是菜单(Menu)的 handle...

## WinMain()和它的4个参数

### WinMain()的类型

介绍完 Instance 与 Handle 之后,我们就来看看 Windows 程序的主体:WinMain()。WinMain()是 Windows 程序的开头,就像一般 C 程序是由 main()开头一样。WinMain()也和一般函数一样有它的类型,其类型声明如下:

```
int PASCAL WinMain()
```

WinMain()的类型是 int,也就是说我们可由程序返回一个整数(int)给调用者,这对程序间的沟通很有帮助。此外,Microsoft 为了提高 Windows 程序的效率,特别规定函数必须是以 PASCAL 的顺序来传递变量,这和一般 C 函数传递变量的顺序正好相反,但速度较快。所以类型声明时要加上 PASCAL 这个修饰词(Modifier)。

### WinMain()的4个参数

WinMain 有 4 个参数列出如下:

| 类型                 | 参数                       |
|--------------------|--------------------------|
| ↓                  | ↓                        |
| int PASCAL WinMain | (HINSTANCE hInstance,    |
|                    | HINSTANCE hPrevInstance, |
|                    | LPSTR LPCmdLine,         |
|                    | int nCmdShow)            |

其中 HINSTANCE、LPSTR、int 为参数的类型,除了 int 为 C 语言的标准类型之外, HINSTANCE 和 LPSTR 都是在 windows.h 文件中定义的。HINSTANCE 即 Instance 的 Handle,而 LPSTR 则为字符串(String)的远程指针(Long Pointer)。

### hInstance 与 hPrevInstance

hInstance 与 hPrevInstance 就是 Windows 传给 WinMain()的两个 Handle。其中:

- ☐ hInstance: 本程序实例(Instance)的句柄(Handle)。每当程序运行时,Windows 会赋与一 handle 给该 Instance,此 handle 会由 WinMain()的 hInstance 参数输入。
- ☐ hPrevInstance: 前一程序实例的句柄。若本程序之前已经运行,并且尚未结束,则前一个