

中国海洋大学教材建设基金资助

Fortran 90 编程基础

主 编 林晓彤

副主编 杜树杰

中国海洋大学出版社

• 青 岛 •

第一章 Fortran 90 和 Visual Fortran 的开发环境

若想让计算机按人的意愿解决问题、处理问题,即实现人所提出的算法,需要进行相应的程序设计,这就需要学习程序设计语言。世界上最早出现并得到广泛应用的高级语言是 Fortran 语言。

Fortran 是 FORmula TRANslation 的缩写,原意为“公式翻译”,最初是为数值计算而设计的。

第一个 Fortran 语言文本是 1954 年提出的,1956 年开始真正使用。随后经不断发展,形成了多种版本。1958 年出现了 Fortran II。1962 年出现 Fortran IV。

1966 年,美国国家标准化协会(ANSI)以 Fortran IV 为基础制定了美国标准文本,简称 Fortran 66。1972 年,国际标准化组织(ISO)将 Fortran 66 作为 ISO 当时的 Fortran 标准文本。

1976 年,ANSI 对 Fortran IV 进行修订,并于 1978 年推出了新的标准 Fortran 77。

1991 年, Fortran 90 问世。

随着面向对象程序设计技术的发展和成熟,可视化编程越来越成为目前发展的主流和趋势,在这个可视化的编程潮流中,许多程序设计语言都得到了蓬勃的发展,像 C 语言发展到了 Visual C++, Basic 语言发展到了 Visual Basic。而作为最早、最广泛使用的、在科学和工程计算领域作出巨大贡献的高级语言, Fortran 语言也从 Fortran 77 发展到了 Fortran 90,借助于新的编译平台, Fortran 90 可以像 VC, VB 一样,实现可视化的编程,还可以实现与其他语言的混合编程。

另外,在 Fortran 90 中增加了许多新的特征和功能,提高了 Fortran 语言的灵活性和使用性,克服了 Fortran 77 的不足。例如,新增加了递归调用和控制语句,使程序设计更加方便;可以设计动态数组、重载函数,提供了指针类型和派生类型,提高了资源的可利用性,也方便了用户。

微软公司专门为 Fortran 90 开发了新的编译开发平台——Microsoft Fortran Powerstation,在此平台上, Fortran 90 可以方便地调试和开发。本书所要介绍的集成开发环境 IDE——Digital Visual Fortran 6.0 就是在它的基础上发展、改进而来的。它的界面与 Microsoft Visual C++ 的集成开发环境基本一致。

下面,我们就简要地介绍一下 Fortran 90 的新特性,以及 Digital Visual Fortran 6.0 开发环境(IDE)的使用方法。

1.1 Fortran 90 的新特性

如果你以前学习或使用过 Fortran 77,就会发现它有很多的不便和不足之处:

1. 没有任何动态存储工具。
2. 没有用户自己定义的数据类型和数据结构。
3. 容易产生一些编译器所检测不到的错误,特别是在调用过程时。调查表明,在 400 万行的 Fortran 程序中,有大约 20%的过程接口是有缺陷的。
4. 由于依赖于开发平台的某些特性,程序并不是完全可移植的。
5. 控制结构匮乏,以至于不得不使用 goto 语句和语句标号,使程序的结构混乱。
6. 源程序必须按固定格式(列格式)书写,变量名要少于 6 个字符,语句要使用大写字母等规定,使程序的编写十分不便。

7. 编写程序时,使用扩展的 Fortran 77 标准,也使程序的可移植性降低。

再来看看 Fortran 90 带给我们的新变化:

1. 它的结构特性更好,时代性更强,新增了许多具有现代特性的语句与项目,程序的易读性与可维护性更好。
2. 自由格式的源代码书写方式,使程序的书写十分方便、灵活。
3. 信息管理功能得到了提高,它引进了字符种别的概念,使 Fortran 能给各国文字(包括汉字)、各种学术符号(如化学符号、数学符号等)赋予不同的种别参数,从而可以对之进行运算。
4. 引进了数组直接运算、数组直接赋值、数组与标量的运算等概念,大大地简化了程序的编写。数组可以作为一个整体,出现在表达式、赋值语句和函数中。
5. 按数据结构需要,设置派生类型、指针以及模块等,可以很方便地操作各种数据结构。允许动态存储分配,借助指针可以构造复杂的动态数据结构。
6. 引入了新的程序单位——模块,可以封装数据和与之相关的一系列过程。
7. 新增加了 select case, cycle, exit 等新的控制结构语句,使程序设计中极少出现跳转语句和标号语句。
8. 允许递归函数的使用;过程调用时可以使用可选项参数调用、关键词调用、类属过程调用等,使得过程间的调用更加灵活。
9. 增加了运算符超载、赋值号超载功能,扩展了 Fortran 语言的操作功能。
10. 语法结构更为严谨规范,从高层语法到低层语法,都有统一的结构,使编程和调用都很规范,也提高了程序的可读性。

Fortran 90 的这些新特性,使程序的编写更简单,程序更容易维护;由于编译

器能发现更多的错误,所以程序更稳定、安全;程序的可移植性增强,这是由于依赖于机器的特性很少存在。在 Fortran 90 中, Fortran 77 是它的子集,所以你可以在编程中逐步地学习和熟悉 Fortran 90 的新特性,完成从 Fortran 77 到 Fortran 90 的过渡。

1.2 Digital Visual Fortran 6.0 系统的安装

通过安装光盘中的安装程序(setup. exe)来安装 Digital Visual Fortran 6.0:

1. 启动 Windows 系统。
2. 将 Digital Visual Fortran 6.0 的安装盘放置到光盘驱动器中。
3. 从桌面上“我的电脑”中找到光驱,双击其中的 setup. exe 文件。
4. 以后的安装按照安装向导的提示操作。

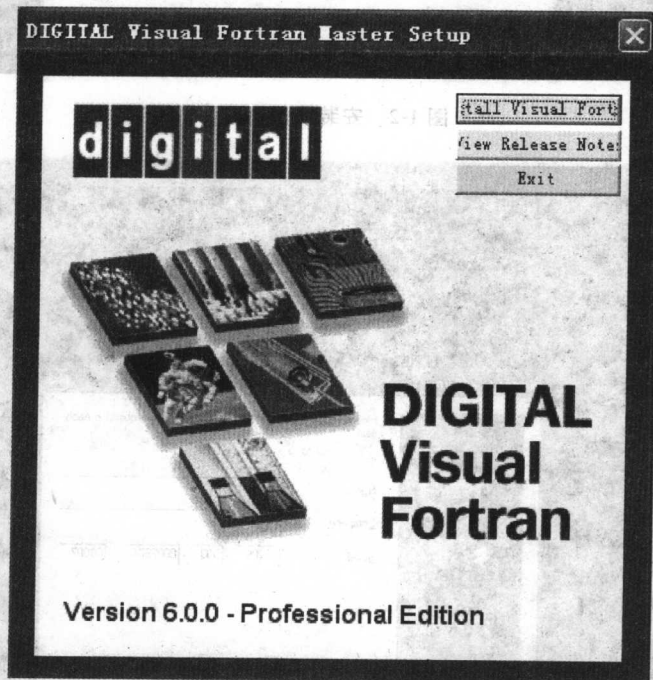


图 1-1 Digital Visual Fortran 安装的确认窗口

第一步,按安装程序的提示,同意接受安装协议,并按要求输入公司(组织)的名称和序列号等信息。

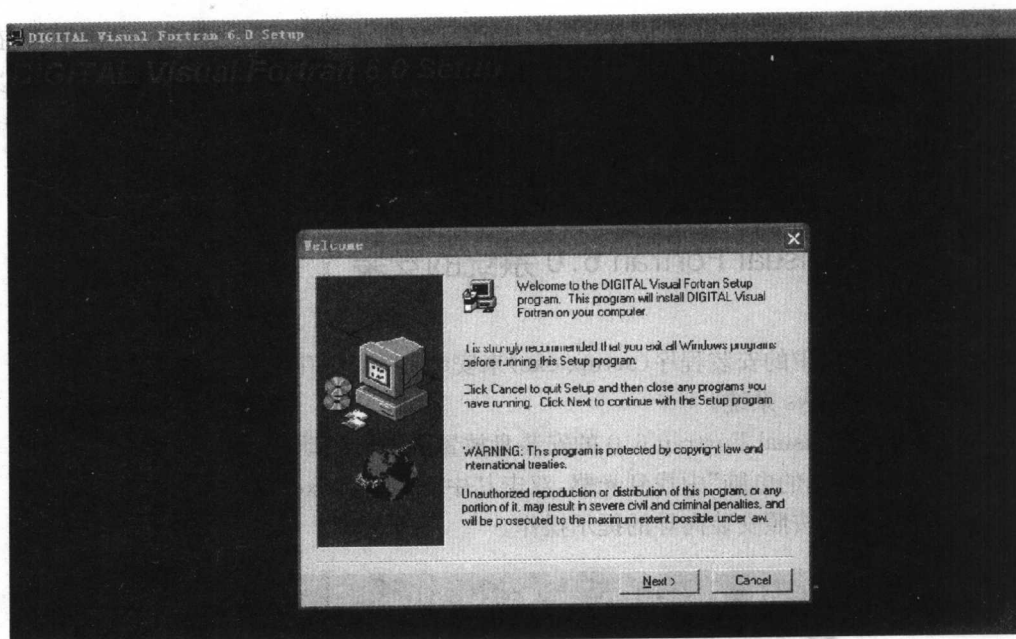


图 1-2 安装协议窗口

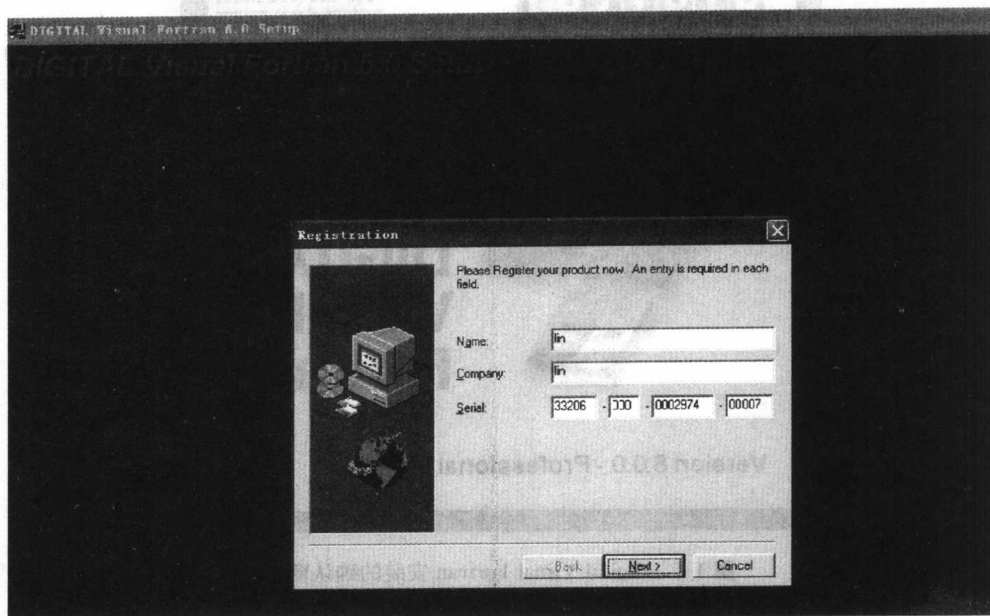


图 1-3 输入序列号

第二步, 安装程序提示你选择如下几种安装方式:

(1) Typical: 典型安装。它是程序内定的安装方式, 安装大部分选项, 占用磁盘空间 191 MB。

(2) Custom: 自定义安装。可以对所安装项作挑选。

(3) Run from CD-ROM: 光盘运行安装。它仅拷贝共享文件到硬盘, 大部分程序仍然在光盘上, 占用磁盘空间 27 MB。

你还可以改变默认的安装路径, 把 Digital Visual Fortran 6.0 安装到其他目录或其他硬盘分区中。

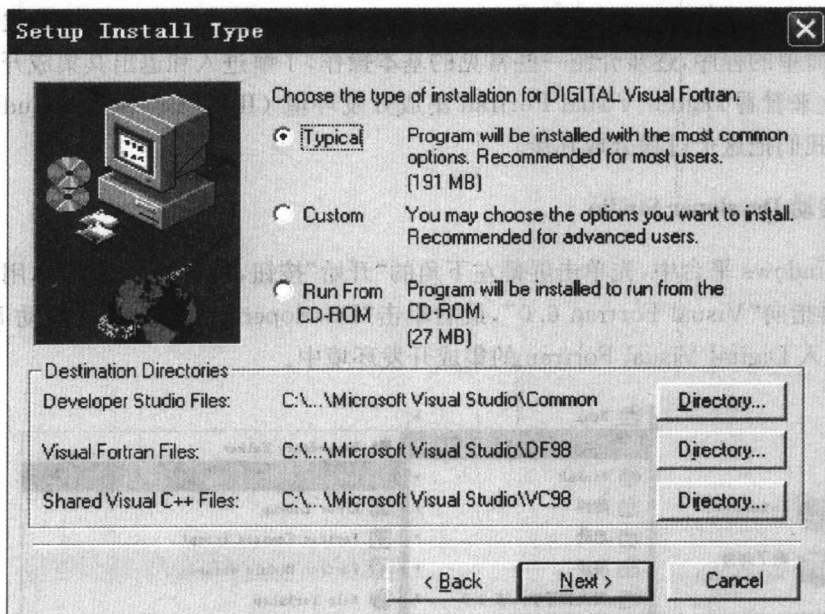


图 1-4 选择安装类型

第三步, 在你选择自己需要的安装方式后, 安装程序会自动地完成其他的安装操作。

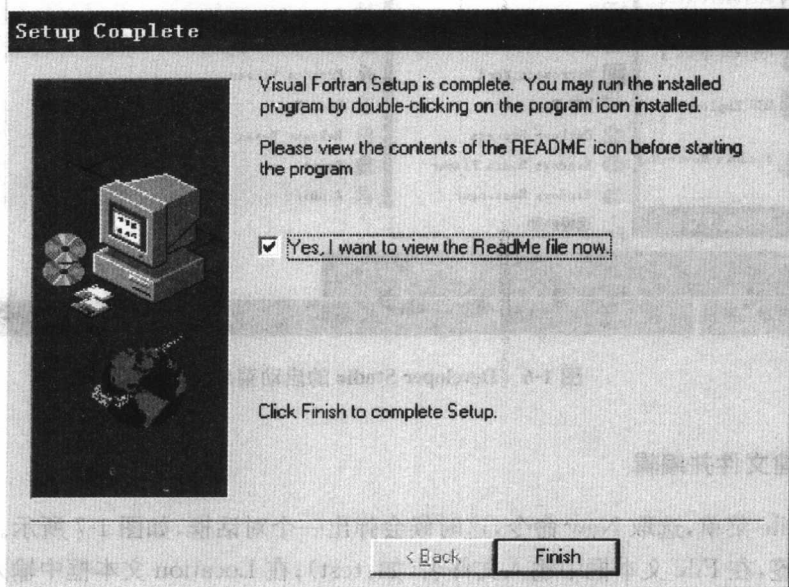


图 1-5 安装完成窗口

1.3 集成开发环境的初步接触

Digital Visual Fortran 为用户提供了一个功能强大的编程平台。下面让我们通过编写一个最简单的程序,逐步介绍一些常见的基本操作,了解进入和退出其集成开发环境的步骤,由此来看看 Digital Visual Fortran 集成开发环境 (IDE) Developer Studio 是怎么工作的。我们把这个过程分成几步。

1.3.1 启动 Developer Studio

在 Windows 平台中,先单击屏幕左下角的“开始”按钮,打开“开始”菜单,用鼠标指向“程序”,再指向“Visual Fortran 6.0”,最后单击“Developer Studio”,从而启动 Developer Studio,进入 Digital Visual Fortran 的集成开发环境中。

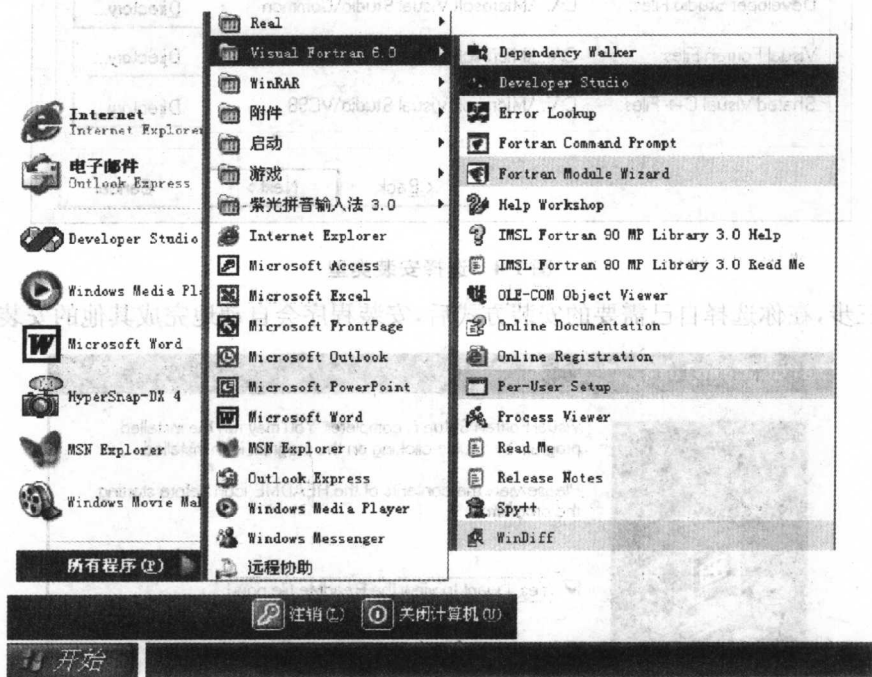


图 1-6 Developer Studio 的启动菜单

1.3.2 新建文件并编辑

打开 File 菜单,选取 New 命令,这时候会弹出一个对话框,如图 1-7 所示。单击其中的 Files 标签,在 File 文本框中输入文件名(如: test);在 Location 文本框中输入文件所要保存的路径(也可以单击旁边的“...”按钮,从中选择保存文件的文件夹);选取文件类型

为“Fortran Free Format Source File”,即 Fortran 90 自由格式源程序,然后单击“OK”按钮,这时右边窗口中就会出现一片空白的正文区等待用户输入。

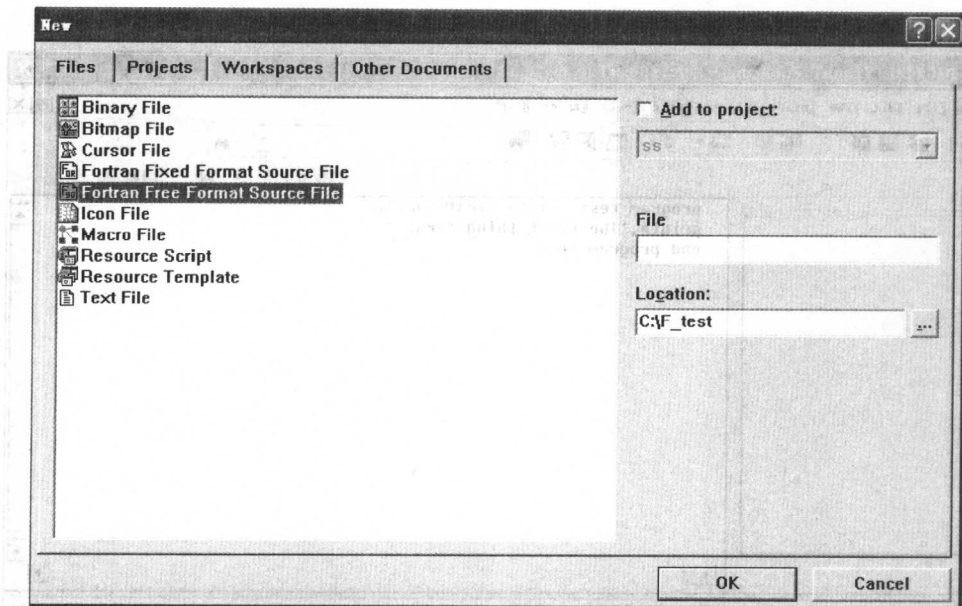


图 1-7 New 对话框的 Files 标签

为将文本添加到文件中,用鼠标点击一下右边窗口中文本将存放的位置,然后就可以输入源程序代码。缺省情况下,编辑器处在“插入状态”,也就是说,新键入的文本将使旧文本向后推移。按 Ins 键,可以切换到“改写状态”。这时,所输入的文本将覆盖掉已存在的旧文本。同时,状态栏显示“OVR”字样,提醒用户正处在改写状态。如果再按一次 Ins 键,将回到插入状态。通过使用鼠标,或者使用光标键,可以在文件中上下移动,进行插入或修改。

与 Word 中的操作一样,文本块选择可用鼠标键单击文本的一端,一直保持鼠标键为按下状态,直到鼠标移到文本的另一端为止才释放鼠标键;选择文本块时同时按下 Alt 键,可以选中列文本块。文本块可以被复制、剪切。

我们在右边编辑窗口中输入下列源程序代码:

```
program test
! the first sample
print *, 'The first thing first'
end program test
```

你一定注意到了出现的代码使用了色彩调配。Developer Studio 用“语法着色”对代码的单元给予高亮显示。缺省情况下,代码为黑色,注释为绿色,关键字是蓝色。

语法着色可以帮助用户发现一些关键字的拼写错误。如 integer 写成 integre, 其颜色就不会呈蓝色, 提醒用户关键字写错, 因而它能在进入编译之前, 防止许多编译错误的发生。

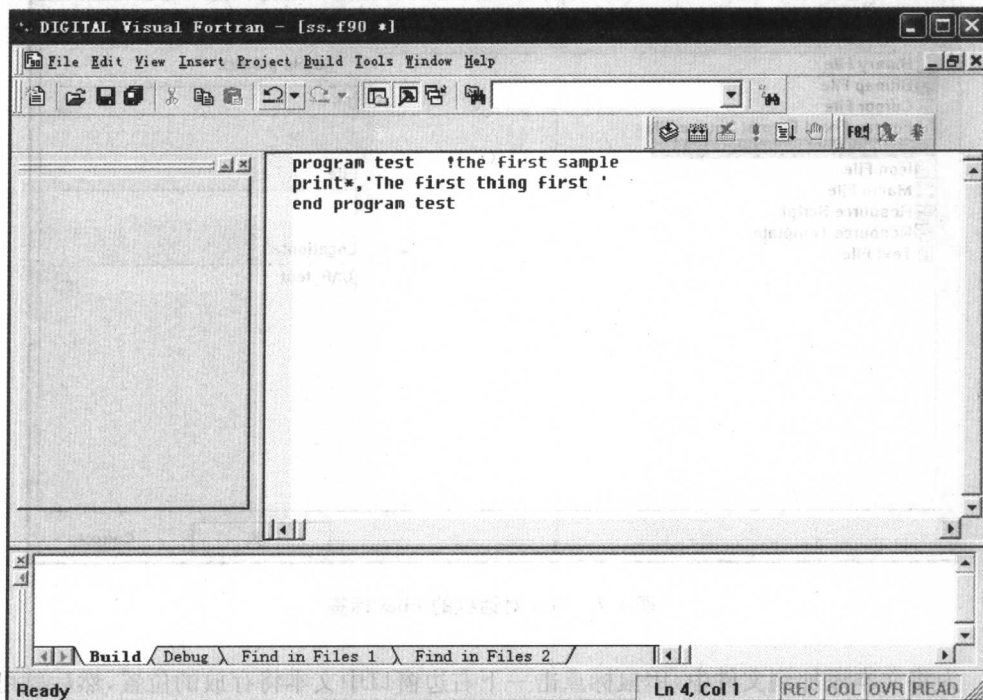


图 1-8 Digital Visual Fortran 的编辑窗口

1.3.3 保存新文件

为了把刚刚输入的文件保存下来, 应打开 File 菜单, 选 Save 命令, 也可以使用 F2 快捷键, 把输入的源程序按你指定的文件名, 保存到指定的位置。

1.3.4 编译和连接文件

打开 Build 菜单, 然后选用 Build $\times\times\times$. exe 菜单项(快捷键 F7), 这时候 Developer Studio 将弹出一个消息框, 告诉用户: 编译连接命令需要有一个激活的工作区, 并询问用户是否想要创建一个缺省的项目工作区。这时选择“是”按钮, 接下来集成开发环境 Developer Studio 就会为 $\times\times\times$. f90 创建一个缺省的项目工作区, 并开始进行编译和连接, 最后, 当输出窗口中出现

```
 $\times\times\times$ . exe—0 error(s), 0 warning(s)
```

字样时, 表示 $\times\times\times$. exe 可执行文件已经生成。

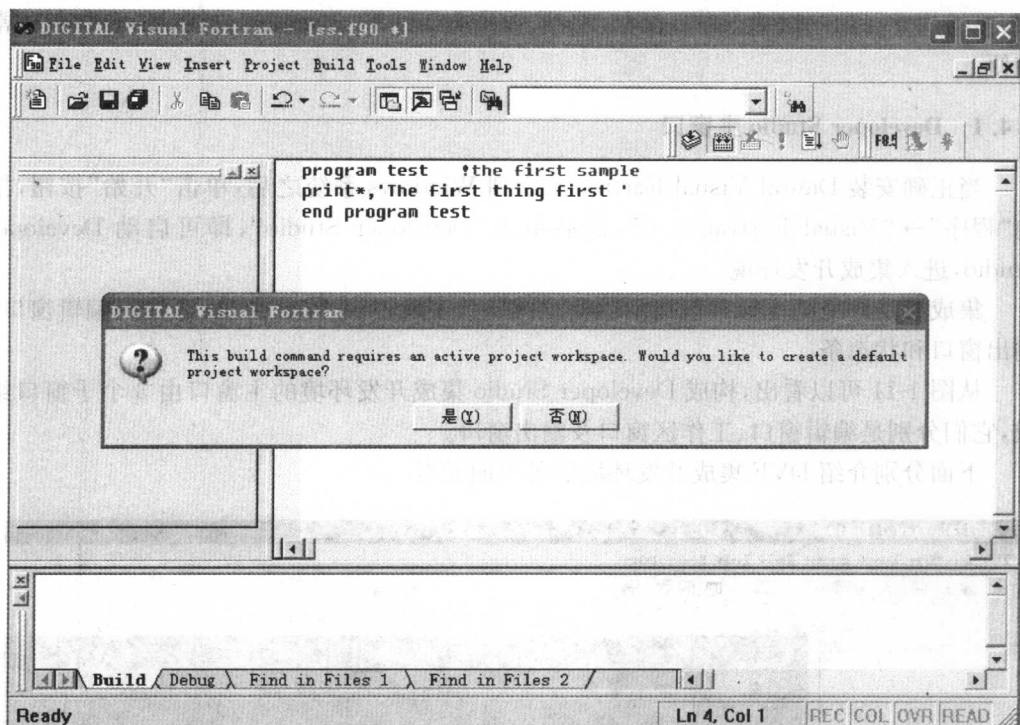


图 1-9 编译连接时建立缺省的项目工作区

1.3.5 运行程序和退出 Developer Studio

在 Build 菜单中选取 Execute $\times\times\times$.exe 命令(快捷键 Ctrl+F5),就可以运行刚刚生成的 $\times\times\times$.exe,其运行结果如图 1-10 所示。

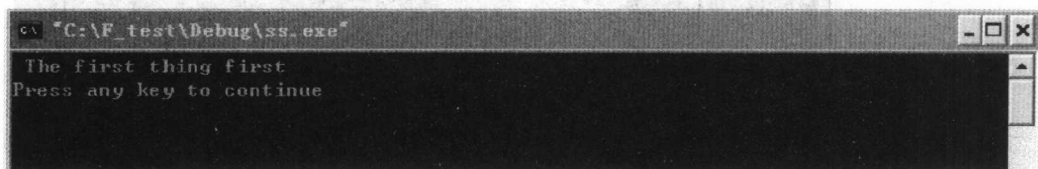


图 1-10 程序的运行结果窗口

在结果窗口中,有一句话“Press any key to continue”,表示程序运行完后,按任意键都将返回 Developer Studio 集成开发环境中。

在 File 菜单中选用 Exit 命令,就会退出 Developer Studio 的集成开发环境。

1.4 Developer Studio 的集成开发环境

Developer Studio 提供了良好的可视化编程环境,使得在应用程序中可进行各种操

作,包括建立、打开、浏览、编辑、保存、编译、连接和调试等,点按鼠标、按钮即可进行,非常方便。

1.4.1 Developer Studio 主窗口

当正确安装 Digital Visual Fortran 6.0 到 Windows 系统之后,单击“开始”按钮,指向“程序”→“Visual Fortran 6.0”,最后单击“Developer Studio”,即可启动 Developer Studio,进入集成开发环境。

集成开发环境的主窗口包括标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态条。

从图 1-11 可以看出,构成 Developer Studio 集成开发环境的主窗口由 3 个子窗口组成,它们分别是编辑窗口、工作区窗口及输出窗口。

下面分别介绍 DVF 集成开发环境的各界面元素。

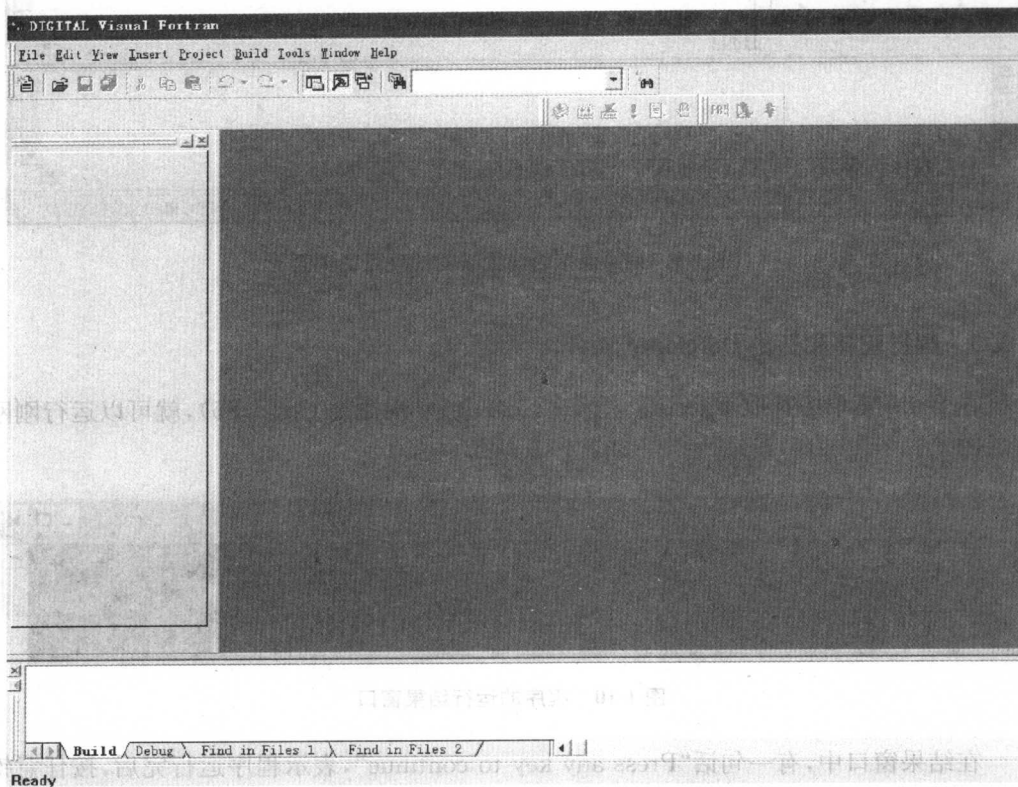


图 1-11 Developer Studio 主窗口

1.4.2 Developer Studio 工具栏

为了使用工具栏,必须使工具栏出现在主窗口中,缺省时已经使得 Standard 和 Build Minibar 两个工具栏显示在主窗口中。要使用其他工具栏,只需移动鼠标指向工具栏、菜

单栏或编辑区的位置,然后单击右键,屏幕上弹出一快捷菜单,单击要使用的工具栏前面的复选框,使其前面出现“√”,则相应的工具栏就会出现在 Developer Studio 主窗口中。

下面逐一介绍常用工具栏所含的各工具按钮的作用与用法。

(一) Standard 工具栏

Standard 工具栏含有 15 个工具按钮,主要用于建立项目工作区及项目,如图 1-12 所示。当鼠标移到某一按钮上时,会显示出该按钮的作用和对应的快捷键。



图 1-12 Standard 工具栏

各按钮的具体功能如下:

New Text File: 创建新的文本文件。

Open: 打开已有的文档。

Save: 保存文档。

Save all: 保存所有打开的文件。

Cut: 剪切选定的内容到剪贴板中。

Copy: 复制选定的内容到剪贴板中。

Paste: 在当前插入点处插入剪贴板中的内容。

Undo: 取消最后的操作。

Redo: 重复先前的操作。

Workspace: 显示或隐藏工作区窗口。

Output: 显示或者隐藏输出窗口。

Windows list: 管理当前打开的窗口。

Find in Files: 在多个文件中搜索字符串。

Find: 激活查找工具。

Search: 搜索联机文档。

(二) Build Minibar 工具栏

该工具栏包含有 6 个工具按钮,如图 1-13 所示。主要用于建立较小的项目。



图 1-13 Build Minibar 工具栏

Compile: 编译文件。要编译的文件必须是构成项目的一部分。

Build: 创建可执行文件。

Stop Build: 停止创建可执行文件。

Execute Program: 执行程序。

Go: 启动或继续程序的执行。

Insert/Remove Breakpoint: 插入或删除断点。

1.4.3 Developer Studio 菜单栏

Developer Studio 菜单栏包括的菜单有 File, Edit, View, Insert, Project, Build, Tools, Window 和 Help 等 9 个主菜单。下面简要介绍一下常用菜单的功能。

File Edit View Insert Project Build Tools Window Help

图 1-14 Developer Studio 菜单栏

(一)File 菜单

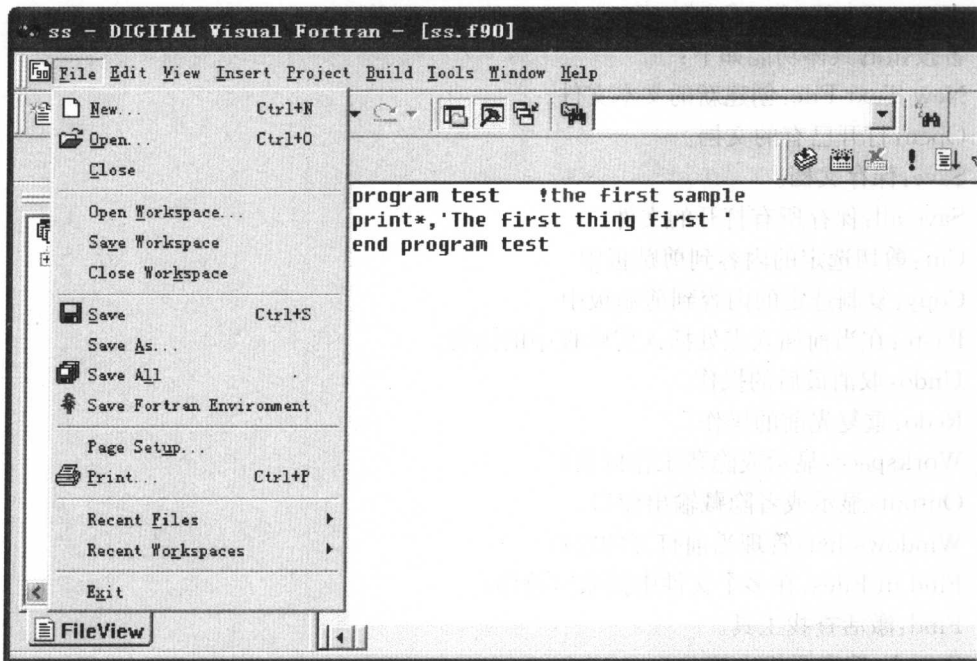


图 1-15 File 菜单栏

File 菜单的命令主要完成文件的建立、保存、打开、关闭以及打印等工作。常用的命令有如下几个。

1. New 命令

选择该命令将打开一个对话框(如图 1-16 所示),通过该对话框用户可以创建新的文件、项目、工作区或者其他文档。该窗口有四个标签页:Files, Projects, Workspaces 和 Other Documents。

缺省时打开 Files 标签,在其中单击要创建的文件类型,然后在 File 文本框中输入要创建的文件的名字。可以创建的文件类型包含有 Fortran 自由格式源文件(Fortran Free Format Source File)和 Fortran 固定格式源文件(Fortran Fixed Format Source File)等。

选择 Project 标签,将打开如图 1-16 所示的 Project 选项卡。单击要创建的项目类

型,然后在 Project name 文本框中键入项目的名字。其中包含 Fortran 控制台应用程序 (Fortran Console Application)。

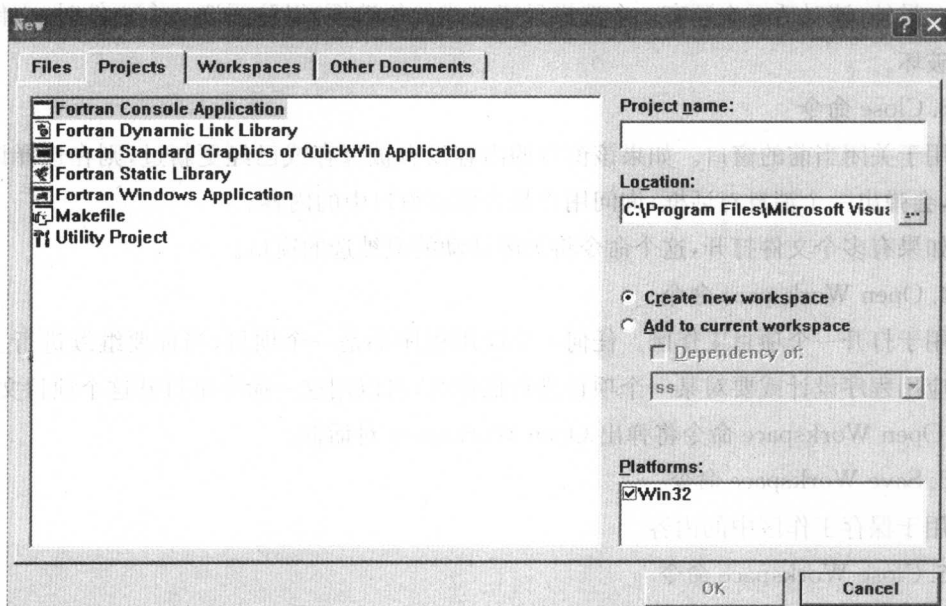


图 1-16 New 对话框的 Projects 标签

在 New 对话框的 Workspaces 和 Other Documents 选项卡中,可以创建各种类型的工作区文件和文档。

2. Open 命令

用于打开一个已经存在的文件。选择该命令弹出一个 Open 对话框窗口,如图 1-17 所示。

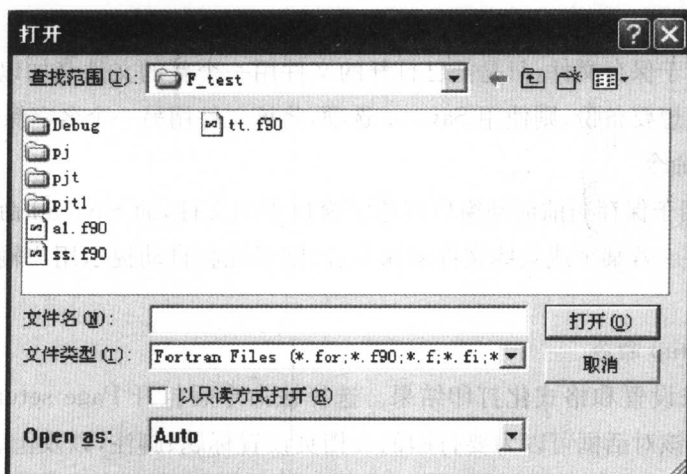


图 1-17 Open 对话框

该对话框可以从已存在的目录和驱动器中选择相应的文件,并且对话框还带有通配符文件名文本框和文件类型的选择框,用于在选择一个文件打开之前选取要查看的文件集合。另外,该对话框中还有一个选择只读方式的复选框,以防阅读一个文件时,文件被意外破坏。

3. Close 命令

用于关闭当前的窗口。如果该窗口的内容没有被保存或已经更新过,则在选择该命令后,会弹出一个消息对话框,询问用户是否保存窗口中的内容。

如果有多个文件打开,这个命令将关闭活动的或被选的窗口。

4. Open Workspace 命令

用于打开一个项目工作区。任何一个应用程序都是一个项目,当你要继续进行上一次的应用程序设计或要对某一个项目进行操作时,可以用这一命令来打开这个项目文件。选择 Open Workspace 命令将弹出 Open Workspace 对话框。

5. Save Workspace 命令

用于保存工作区中的内容。

6. Close Workspace 命令

用于关闭项目工作区。当项目文件的内容没有被保存或已经更新过时,选择该命令会弹出一个消息对话框,询问是否保存所有的内容。

7. Save 命令

用于保存活动的窗口或当前选定窗口中的文件内容。若所保存的文件是首次保存,则系统会提示 Save as 对话框;如果当前文件是只读方式打开的,则 Save 选项是灰色的,表示此项不能使用,即不能修改该文件。

8. Save as 命令

该命令也用于保存文件,只是把已打开的文件用一个新的文件名加以保存而已。如果在保存文件时想要备份,则使用 Save as 选项,将该文件用另一个名字保存起来。

9. Save all 命令

Save 命令用于保存当前活动窗口或选定窗口中的文件,而 Save all 命令用来保存所有窗口内的文件。若某个或某些文件未保存过,则系统会自动提示用户输入有效的文件名。

10. Page setup 命令

该命令用来设置和格式化打印结果。选择该命令后打开 Page setup 对话框,如图 1-18 所示。利用该对话框可以为要打印的文档页设置标题、脚注,以及上、下、左、右 4 个边距。

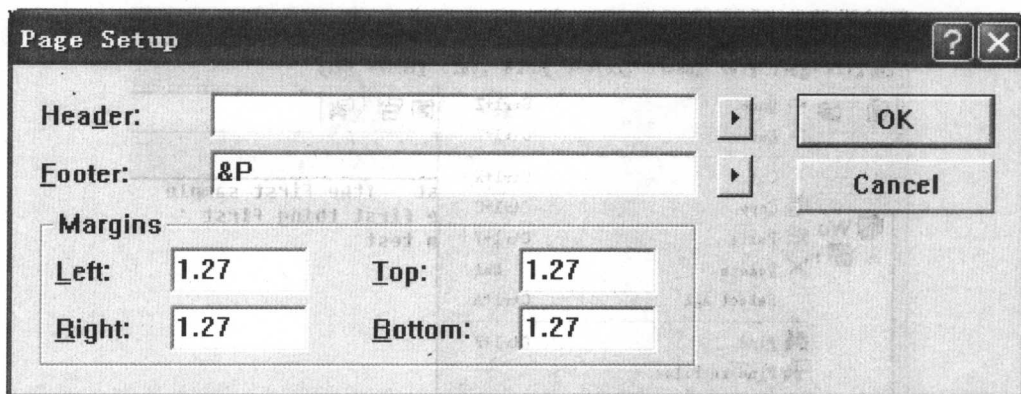


图 1-18 Page Setup 对话框

11. Print 命令

用于打印全部或所选的当前被编辑的窗口的内容。选择这一命令将弹出一个对话框,从中可以选择打印机类型、要打印的文本部分和设置打印机等。

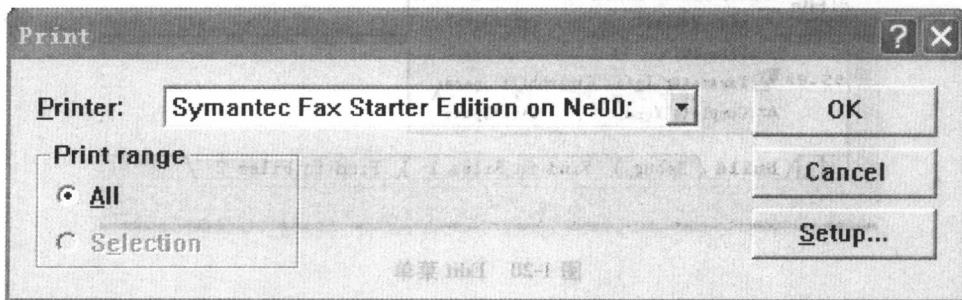


图 1-19 Print 对话框

12. Recent files 命令

列出最近的四个被编辑过的文件。通过这些列表文件,可以方便地找到最近编辑过的文件,而不必通过驱动器和文件夹去查找,从而节省很多时间。

当第一次使用时,这一部分是空白的,因为没有被打开的文件可以作为历史记录。

13. Recent workspaces 命令

和“Recent files”命令功能相似,显示最近打开的四个项目工作区。

14. Exit 命令

使用 Exit 命令可退出 Digital Visual Fortran 开发环境。如果所作的修改未被保存,系统在退出之前会询问是否保存所作的更改。

(二) Edit 菜单

1. Undo 命令

用于取消你最近所做的编辑任务。

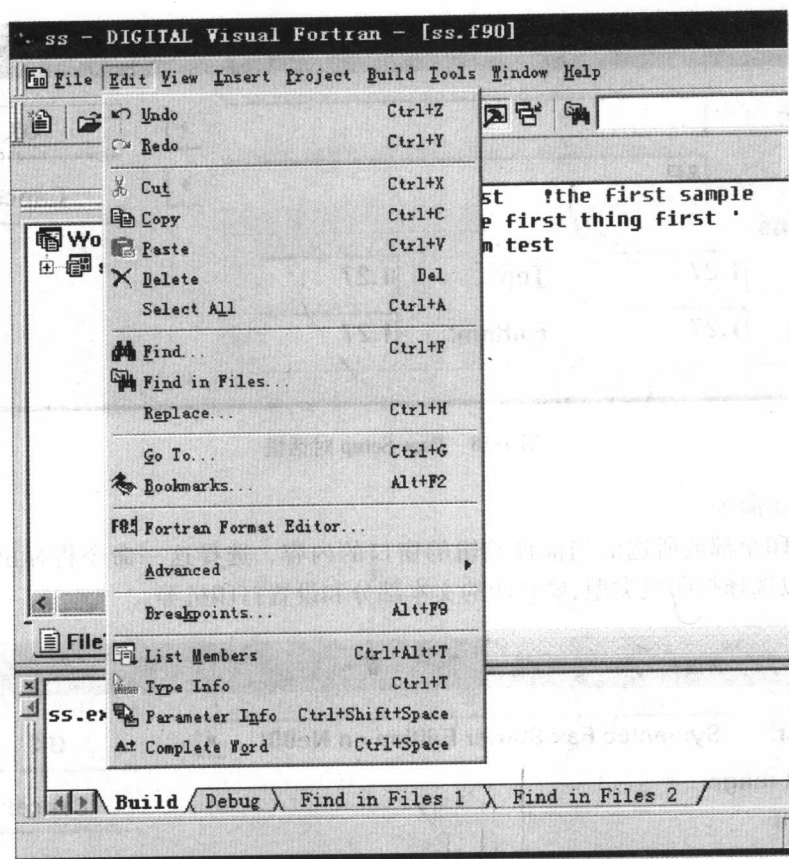


图 1-20 Edit 菜单

2. Redo 命令

用于恢复 Undo 所做的工作,通常使用这项命令来再次执行一个原来认为是错误的有效编辑命令。如果过多地使用了 Undo 命令,Redo 命令只能恢复其中的一部分。

3. Cut 命令

用于剪切被选择的文本,并且把这部分文本拷贝到剪贴板上,覆盖掉剪贴板上原有的内容。

4. Copy 命令

用于将被选择的文本拷贝到剪贴板上,覆盖掉剪贴板上原有的内容。

5. Paste 命令

用于将剪贴板上的内容插入到光标所在的位置。一般使用 Cut 和 Paste 把文本从一个地方移到另一个地方;使用 Copy 和 Paste 把文本复制到另一个地方。

6. Delete 命令

用于删除被选择的文本或光标后面的字符。

7. Find 命令

用于在当前被编辑的窗口中寻找指定的文本。