



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

配置开发环境

- ◆ 1、开发环境的路径参数.....●
- ◆ 2、开发环境的配置.....●

第13章 高级编程技术

- ▶ 实现应用程序开发，除需要大量C语言编程训练、掌握必要的算法和数据结构理论知识之外，还要学习高级编程技术。
- ▶ 高级编程技术有很强的专业性，涉及内容广泛。通常可以分为系统编程、界面编程、多媒体编程、网络编程、数据库编程、硬件编程等六个领域。
- ▶ 在C语言发展进程中，无数科学家和程序员开发出数量众多、性能卓越的专业函数库，充分利用好这些函数库是编写应用程序的重要技巧。

13.1 配置开发环境

- ▶ C语言开发工具安装后，一般会自动配置好标准函数库的使用环境。因此使用标准库函数的C程序，可以直接编译和连接。但如果是使用其他的专业函数库，如GSL科学计算函数库，就需要配置开发环境。

13.1.1 开发环境的路径参数

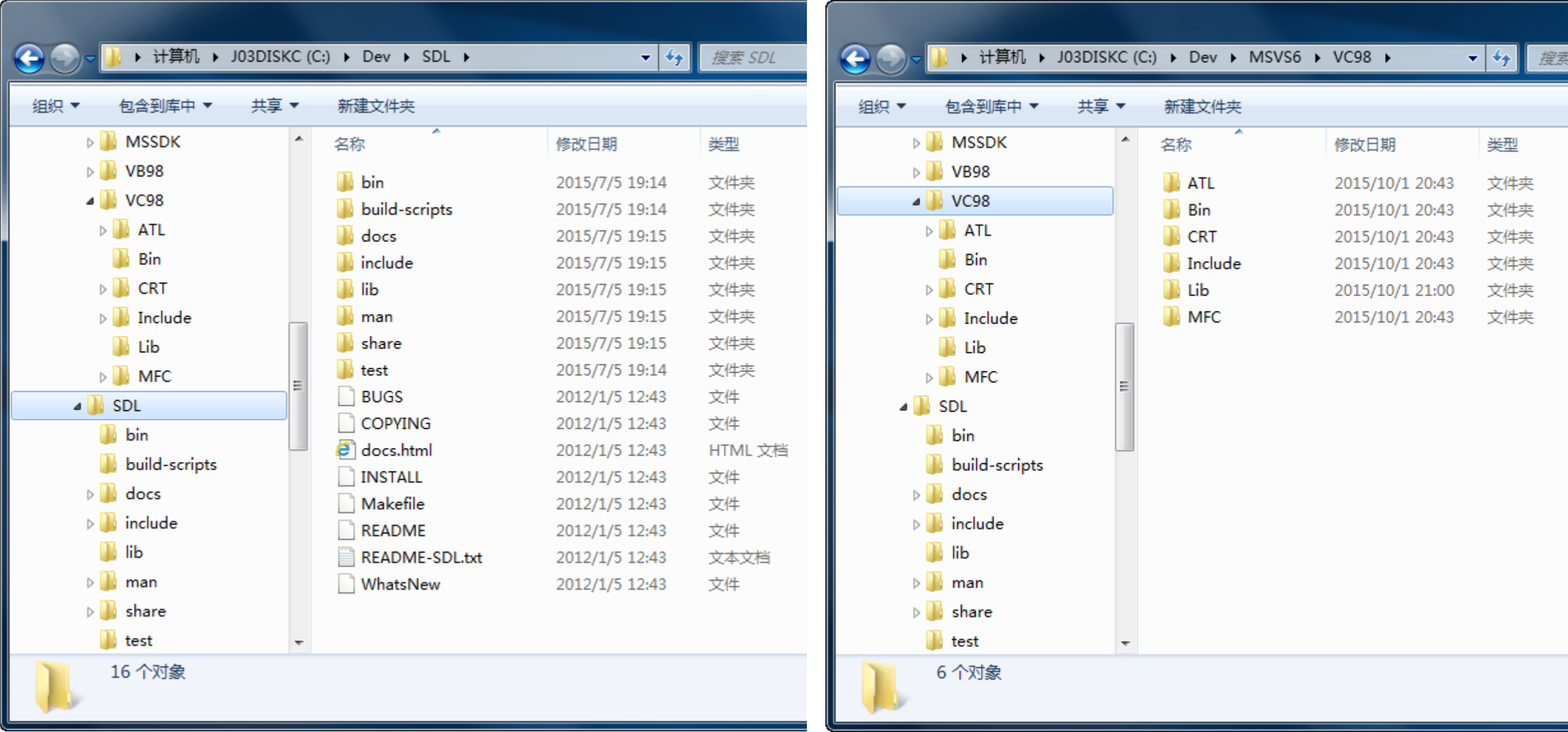
- ▶ 本节针对VC和CodeBlocks，介绍在Windows系统下专业函数库的配置。

13.1.1 开发环境的路径参数

- ▶ 假设VC安装文件夹为<VC>（如C:\DEV\MSVS6\VC98），CodeBlocks安装文件夹为<CodeBlocks>（如C:\DEV\CodeBlocks\MinGW）。由于需要经常使用开发工具的命令行实用程序，所以安装时文件夹名称最好不要有空格，比如像“C:\Program Files”。

13.1.1 开发环境的路径参数

图13.1 Visual C++和CodeBlocks安装文件夹目录结构



13.1.1 开发环境的路径参数

- ▶ 可以看到在<VC>和<CodeBlocks>文件夹中都有include和lib两个文件夹，大多数C语言开发工具基本如此。

13.1.1 开发环境的路径参数

- ▶ 1. 系统INCLUDE路径
- ▶ include文件夹存放开发工具默认安装的全部头文件，称为**系统INCLUDE路径**。可以在这个文件夹中找到我们所熟悉的stdio.h和math.h文件。系统INCLUDE路径是开发工具查找头文件的基础路径，例如math.h头文件包含命令：

```
#include <math.h>
```

13.1.1 开发环境的路径参数

- ▶ 尖括号内的头文件使用相对路径写法（相对于系统INCLUDE路径）。由于math.h在系统INCLUDE路径中，所以文件名前面不用写路径。
- ▶ 如果程序中欲包含<VC>\include\gl\gl.h头文件，由于gl.h文件是在系统INCLUDE路径的gl子文件夹中，因此文件包含命令应为：

```
#include <gl/gl.h>
```

13.1.1 开发环境的路径参数

- ▶ 2. 系统LIB路径
- ▶ lib文件夹存放开发工具默认安装的全部库文件，称为**系统LIB路径**。VC的库文件扩展名为.lib，GCC的库文件扩展名为.a。系统LIB路径是开发工具连接库文件的基础路径，同样使用相对路径来查找库文件。

13.1.1 开发环境的路径参数

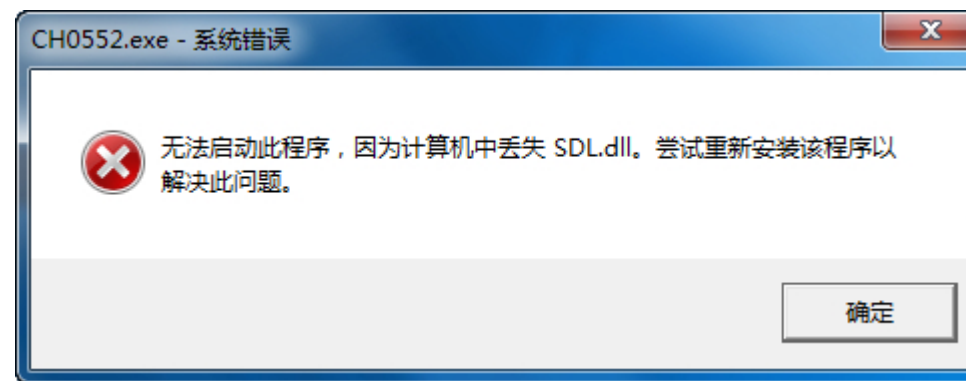
- ▶ 3. 搜索路径
- ▶ 在Windows操作系统中，在“控制面板/系统/高级/环境变量”中有一项“PATH”参数，它是Windows的系统PATH路径。当一个程序运行时，如果需要用到附加的动态链接库DLL（dynamic linking library）文件，则Windows规定该DLL文件要么与程序同处一个文件夹，要么在系统PATH路径中，否则程序不能运行。我们将程序所处的文件夹和PATH路径统称为搜索路径，即查找运行时文件的路径范围，它可以是多个文件夹的集合。

13.1.1 开发环境的路径参数

- ▶ 显然，通过编辑系统PATH路径，可以更改搜索路径。一般地，Windows系统文件夹（如C:\WINDOWS或C:\WINDOWS\system32）是搜索路径之一。如果将DLL文件复制到这个文件夹中，程序运行时能够找到它。

13.1.1 开发环境的路径参数

- ▶ 运行程序时，若出现下面类似提示，则说明缺少附加的动态链接库DLL文件。需要处理搜索路径方可解决问题。



13.1.2 开发环境的路径设置

- ▶ 开发工具的图形界面通常允许程序员设置系统INCLUDE路径和系统LIB路径。

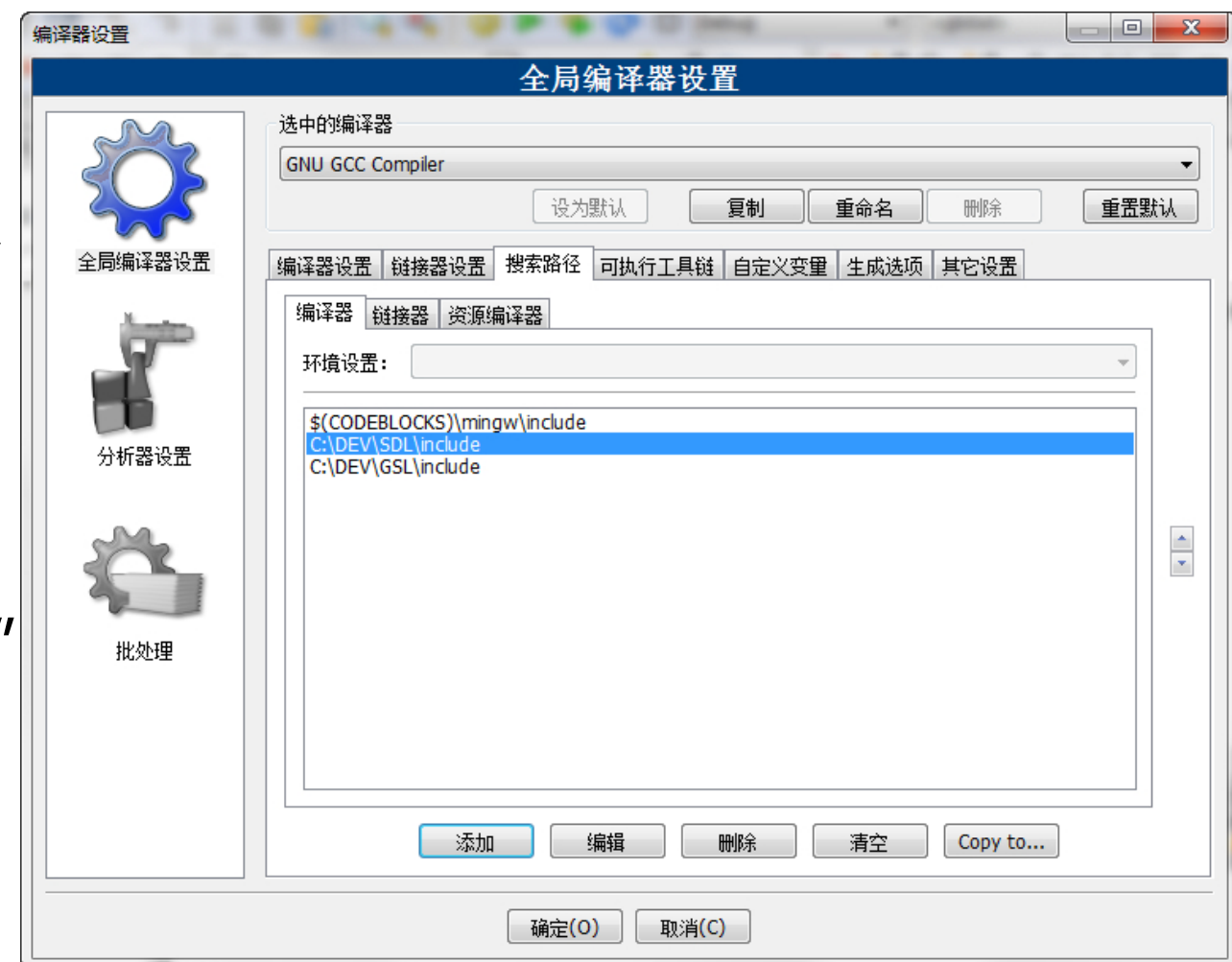
13.1.2 开发环境的路径设置

- ▶ 在CodeBlocks中，单击“Settings/Compiler and debugger...”（设置/编译器和调试器...）菜单，打开“Compiler and debugger settings”（编译器和调试器设置）对话框。

13.1.2 开发环境的路径设置

CodeBlocks的系统INCLUDE路径和系统LIB路径

- ▶ 单击“Search directories”（搜索路径）标签，选择“Compiler或Linker”（编译器或链接器）可以设置CodeBlocks的系统INCLUDE路径和系统LIB路径。在“Search directories”标签中可以任意添加、删除、修改CodeBlocks的系统INCLUDE路径和系统LIB路径。



13.1.2 开发环境的路径设置

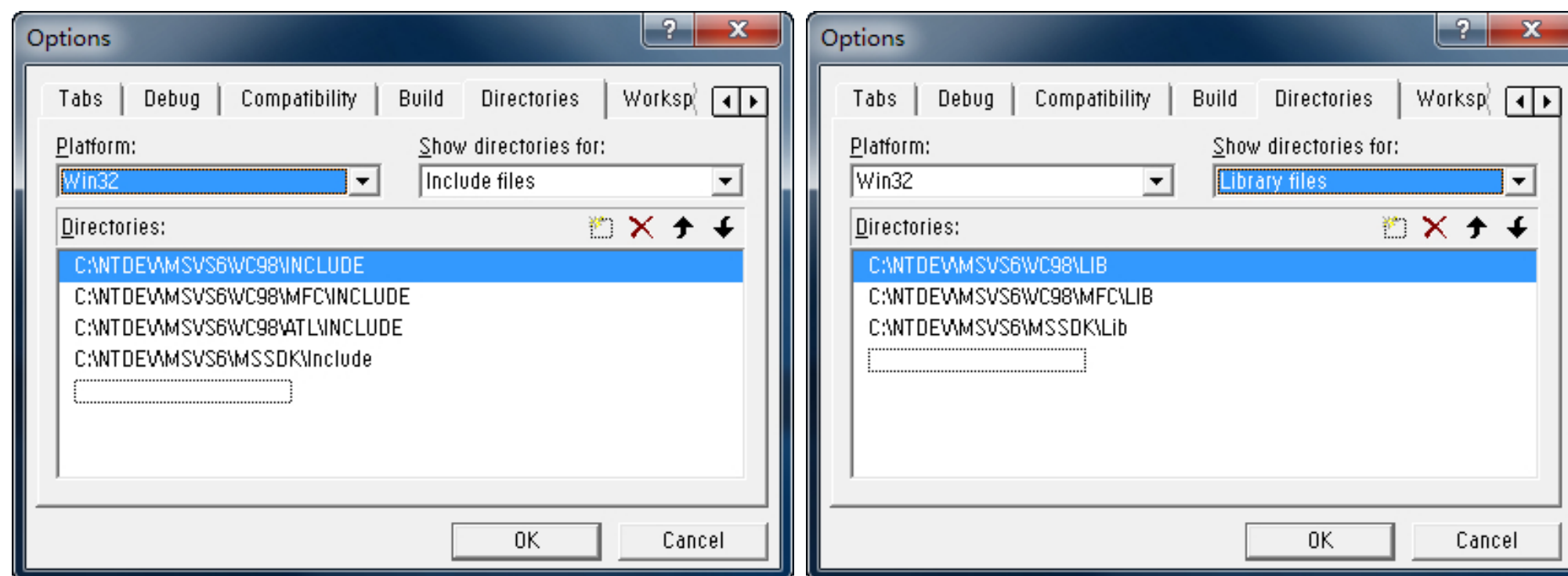
- ▶ 上述路径设置对开发工具而言是全局性的，即路径设置对每个程序和项目工程都是有效的。

13.1.2 开发环境的路径设置

- ▶ 在VC中，单击“Tools/Options...”菜单打开“Options”对话框。单击“Directories”标签，选择“Show directories for:”可以看到VC的系统INCLUDE路径和系统LIB路径。
- ▶ 图中显示VC的系统INCLUDE路径有两个：
“C:\DEV\MSVS6\VC98\include”和“C:\DEV\MSSDK6\include”。
- ▶ 在“Directories”标签中可以任意添加、删除、修改VC的系统INCLUDE路径和系统LIB路径。

13.1.2 开发环境的路径设置

图13.3 Visual C++的系统INCLUDE路径和系统LIB路径



13.1.3 开发环境的配置

- ▶ 通常，一个提供给程序员使用的专业函数库有四部分内容：
 - ①头文件（header）；
 - ②库文件（library）；
 - ③运行时文件（runtime）；
 - ④使用手册文档。

13.1.3 开发环境的配置

- ▶ 头文件的作用是让程序有函数原型、数据结构、宏常量等声明。一般情况下，编译程序时如果出现类似“某某函数未定义或数据未定义”的错误，说明专业函数库的头文件路径没有配置好。

13.1.3 开发环境的配置

- ▶ 库文件的作用是让程序能够连接函数的二进制目标代码。一般情况下，连接程序时如果出现类似“某某函数未找到”的错误，说明专业函数库的库文件路径没有配置好。

13.1.3 开发环境的配置

- ▶ 有的专业函数库需要运行时文件（如动态链接库），通常要将这些文件存放到搜索路径上，使用专业函数库的程序运行时才能找到它们。典型的做法是将这些文件复制到Windows系统文件夹中。

13.1.3 开发环境的配置

- ▶ 专业函数库的使用手册文档使程序员能够正确调用库函数，包含函数原型、函数功能、参数要求、数据含义等说明。

13.1.3 开发环境的配置

- ▶ 专业函数库头文件和库文件的配置一般有三种方法。
- ▶ (1) 复制文件法
- ▶ 可以将专业函数库的头文件和库文件全部复制到开发工具的系统INCLUDE路径和系统LIB路径中，例如<VC>\include和<VC>\lib文件夹、<CodeBlocks>\include和<CodeBlocks>\lib文件夹。如果专业函数库的头文件比较多或者为了与标准库的头文件存放时有所区别，可以将头文件复制到系统INCLUDE路径的一个子文件夹中，但此时#include命令要包含相应的子文件夹路径。

13.1.3 开发环境的配置

- ▶ 复制文件的方法操作简单，操作一次后使得开发工具持久支持专业函数库，如同标准函数库一般。该方法的缺点是开发工具系统路径中的文件会越来越多，而且专业函数库新版本的升级控制有一定的难度，各版本之间的文件容易混淆，给库连接带来潜在冲突。

13.1.3 开发环境的配置

- ▶ (2) 添加路径法
- ▶ 利用开发工具的图形界面可以将专业函数库头文件和库文件所处的文件夹分别添加到开发工具的系统INCLUDE路径和系统LIB路径中。
- ▶ 添加路径的方法操作简单，操作一次后也使得开发工具持久支持专业函数库，而且有便于控制函数库版本升级的优点。当获得了专业函数库新版本时，只需要将原先添加的路径指向新版本的文件夹路径即可。

13.1.3 开发环境的配置

- ▶ (3) 搜索路径法
- ▶ 如果使用开发工具命令行版本的编译程序和连接程序，例如VC的“CL.exe”和CodeBlocks的“gcc.exe”，可以用搜索路径来设置系统INCLUDE路径和系统LIB路径。方法是在Windows“控制面板/系统/高级/环境变量”中设置环境变量参数，表13-1为VC和GCC对应的环境变量参数。
- ▶ 当使用命令行开发工具进行C程序编译和连接时，会将环境变量“INCLUDE”和“LIB”作为查找路径去搜索头文件和库文件。

13.1.3 开发环境的配置

表13-1 开发工具环境变量参数

类别	Visual C++	Code::Blocks
系统INCLUDE路径	INCLUDE	C_INCLUDE_PATH（C语言）， CPLUS_INCLUDE_PATH（C++）
系统LIB路径	LIB	LIBRARY_PATH

13.1.3 开发环境的配置

- ▶ 此外，命令行“CL.exe”和“gcc.exe”程序本身有定位头文件和库文件的命令参数，形式分别如下：

>CL /I dir	指明编译时头文件的查找文件夹dir
>CL /link libfile	指明连接库libfile
>gcc -I dir	指明编译时头文件的查找文件夹dir
>gcc -L dir	指明库文件的查找文件夹dir
>gcc -llibfile	指明连接库libfile

- ▶ 具体细节可以参考CL和gcc的使用手册。

13.1.3 开发环境的配置

- ▶ 设置搜索路径的方法操作繁琐，一般适用于命令行编译和连接方式，初学程序者掌握起来有一定难度。本质上，所有的开发工具都使用命令行方式编译和连接程序，如VC或者CodeBlocks，它们的图形界面只不过是命令行方式的一个“外壳”。

CP 程序设计