



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、第三方开发库的包含和连接.....
- ◆ 2、第三方开发库的配置方法.....

13.1.4 函数库的包含和连接

- ▶ 应用程序包含库是指包含它的头文件，使用#include预处理命令，例如：

```
#include <gsl\gsl_linalg.h> //GSL线性代数函数
```

- ▶ 包含库的头文件后，程序就可以调用库函数和使用它的数据类型、数据结构、宏定义和特殊常量等。

13.1.4 函数库的包含和连接

- ▶ 应用程序连接库是指在程序可执行文件中嵌入库的二进制目标代码或导入库（import library）。
- ▶ 嵌入目标代码方式称为静态连接，它将库的目标代码嵌入到程序可执行文件中。其优点是可执行文件包含库的实现代码，不需要额外的运行时文件就能够运行；缺点是可执行文件体积增大，库代码不能被共享使用。

13.1.4 函数库的包含和连接

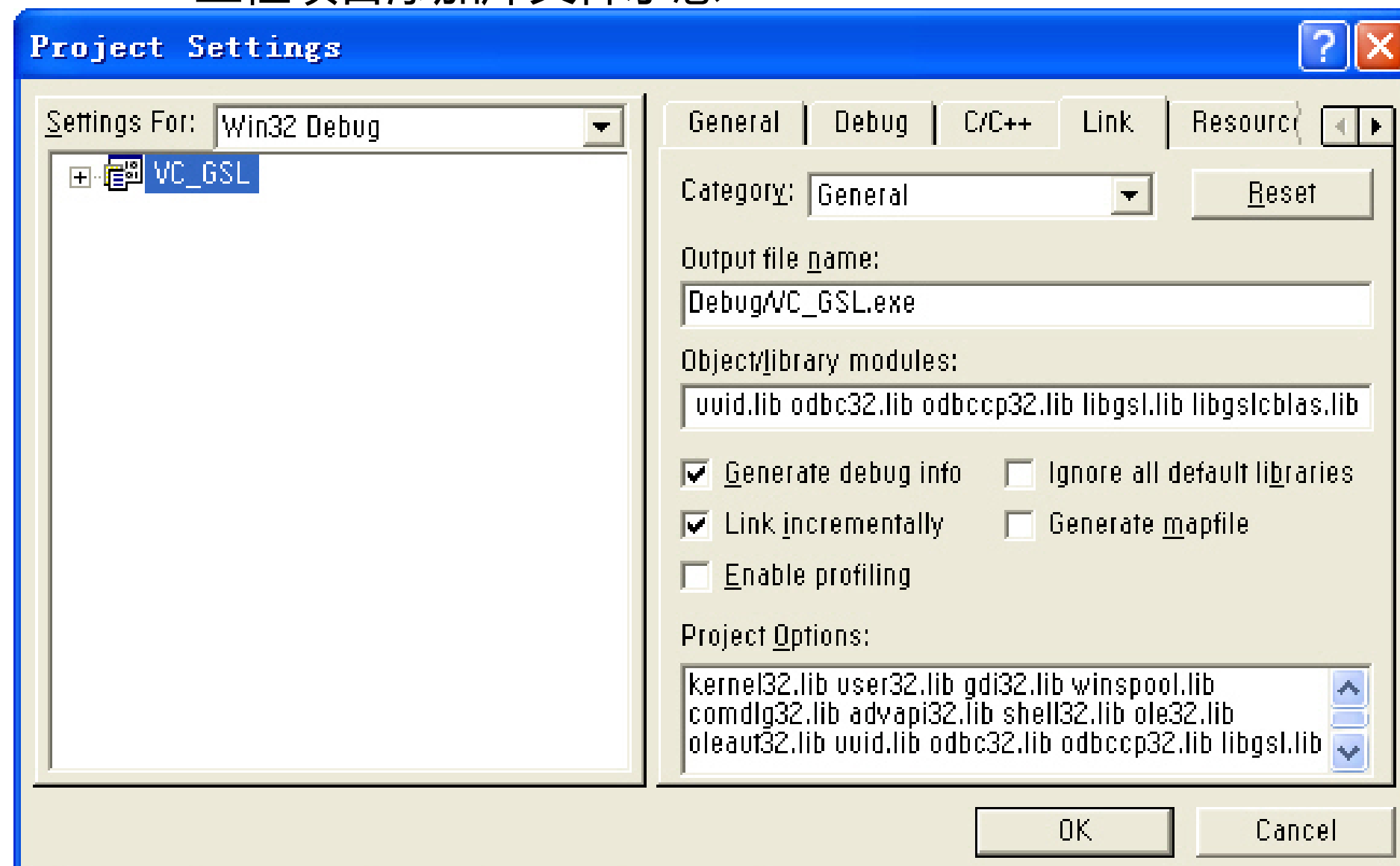
- ▶ 应用程序连接库是指在程序可执行文件中嵌入库的二进制目标代码或导入库（import library）。
- ▶ 导入库方式称为动态链接，它仅将库调用信息存放在可执行文件中，而库代码存放在动态链接库DLL文件中。其优点是可执行文件体积小，一个DLL可以被多个应用程序动态加载共享使用；缺点是程序运行时必须有DLL文件（即在搜索路径中能找到），否则就不能运行。

13.1.4 函数库的包含和连接

- ▶ 在程序连接阶段，专业函数库的库文件 (*.lib或*.a) 必须添加到开发环境中。
- ▶ 在VC中添加库文件的方法是在“Workspace”程序项目的右键菜单中选择“Settings...”菜单命令，或者单击主菜单“Project/Settings...”，打开“Project Settings”对话框，如图所示。在对话框中单击“Link”标签，在“Object/library modules:”后面增加库文件，例如libgsl.lib和libgslcblas.lib，使用空格分隔多个库文件，单击“OK”确定添加、删除、修改库文件操作。这样的操作方式称为VC程序项目添加库。

13.1.4 函数库的包含和连接

图13.4 Visual C++ 工程项目添加库文件示意



13.1.4 函数库的包含和连接

- ▶ VC还提供了另一种等效的添加方式，操作更简单。在源程序文件中，编写连接库预处理命令，形式如下：

```
#pragma comment(lib,"库文件名") //VC连接库文件
```

- ▶ 其中库文件名允许包含绝对路径或相对路径（相对于VC系统LIB路径）。例如：

```
#pragma comment(lib,"libgsl.lib")//VC连接GSL函数库  
#pragma comment(lib,"libgslcblas.lib")//VC连接GSL基础线性代数库
```

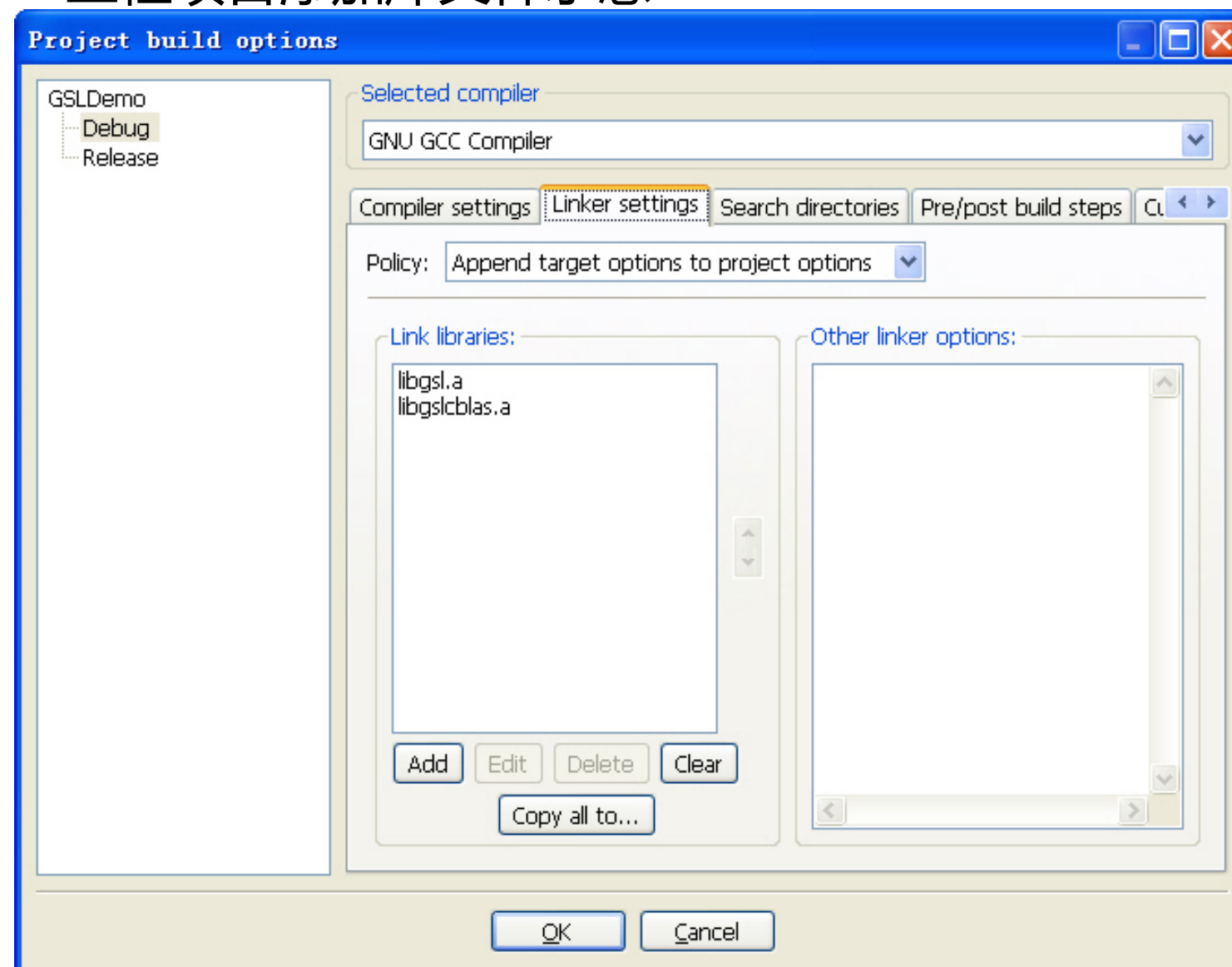
- ▶ 这种方式称为VC连接库预处理。

13.1.4 函数库的包含和连接

- ▶ 在CodeBlocks添加库文件的方法是在“Workspace”（工作空间）右键单击程序项目（project），在弹出菜单中选择“Build options”（构建选项）菜单命令。如图所示，在“Project build options”（项目build选项）对话框中单击“Linker settings”（连接器设置）标签，单击“Add”（添加）增加库文件，例如libgsl.a和libgslcblas.a，单击“Edit”（编辑）修改、单击“Delete”（删除）删除库文件。
- ▶ CodeBlocks不支持VC的连接库预处理命令。

13.1.4 函数库的包含和连接

图13.5 CodeBlocks工程项目添加库文件示意



13.1.5 函数库配置举例

- ▶ 以GSL为例，介绍专业函数库的配置。
- ▶ GNU科学计算函数库GSL（GNU scientific library）是一个强大的C/C++数值计算函数库，是开源自由软件。GSL提供了大量的数值计算函数，如线性代数、特殊函数等等。整个函数库大约有1000多个函数，涵盖了科学计算的各个方面。如表13-2所列。

13.1.5 函数库配置举例

表13-2 GSL函数库目录及对应的头文件

函数库名称	头文件	函数库名称	头文件
异常错误捕获	gsl_errno.h	数学函数	gsl_math.h
复数	gsl_complex_math.h	多项式	gsl_poly.h
特殊函数	gsl_sf.h 或 每个函数各自的头文件	向量和矩阵	gsl_block.h gsl_vector.h gsl_matrix.h
排列	gsl_permutation.h	组合	gsl_combination.h
集合	gsl_multiset.h	排序	gsl_sort.h gsl_sort_vector.h
BLAS	gsl_blas.h	CBLAS	gsl_cblas.h
线性代数	gsl_linalg.h	特征值	gsl_eigen.h

13.1.5 函数库配置举例

续表13-2 GSL函数库目录及对应的头文件

函数库名称	头文件	函数库名称	头文件
N元组	gsl_ntuple.h	数值积分	gsl_integration.h
随机数发生器	gsl_rng.h	拟随机序列	gsl_qrng.h
随机数的分布	gsl_randist.h gsl_cdf.h	统计	gsl_statistics_double.h gsl_statistics_int.h
直方图	gsl_histogram.h gsl_histogram2d.h	快速傅立叶变换	gsl_fft_complex.h gsl_fft_real.h
常微分方程	gsl_odeiv.h	序列加速	gsl_sum.h
数值微分	gsl_deriv.h	Chebyshev近似	gsl_chebyshev.h
插值	gsl_interp.h gsl_spline.h	小波转换	gsl_wavelet.h gsl_wavelet2d.h

13.1.5 函数库配置举例

续表13-2 GSL函数库目录及对应的头文件

函数库名称	头文件	函数库名称	头文件
离散Hankel变换	gsl_dht.h	一维求根	gsl_roots.h
一维最小化	gsl_min.h	多维求根	gsl_multiroots.h
多维最小化	gsl_multimin.h	最小二乘拟合	gsl_fit.h
非线性最小二乘拟合	sl_multifit_nlin.h	基础样条	gsl_bspline.h
蒙特卡洛积分	gsl_monte_plain.h gsl_monte_miser.h gsl_monte_vegas	物理常数	gsl_const_mkssa.h gsl_const_cgsm.h gsl_const_num.h
模拟退火	gsl_siman.h	IEEE浮点算法	gsl_ieee_utils.h

13.1.5 函数库配置举例

- ▶ 在Windows操作系统上使用GSL，首先需要从互联网（<http://www.gnu.org/software/gsl/>）上下载“GSL for Windows”版本，将下载文件解压到临时文件夹TEMP中；然后将TEMP\BIN文件夹中的两个动态链接库libgsl.dll和libgslcblas.dll文件复制到Windows系统文件夹下，使用GSL的应用程序运行时需要这两个文件。

13.1.5 函数库配置举例

- ▶ 1. 在Code::Blocks环境下使用GSL
- ▶ 由于GSL提供的是GCC形式的库文件，而CodeBlocks环境默认使用GCC编译器，因此在CodeBlocks中可以直接使用GSL。
- ▶ 使用前面的“复制文件法”，将TEMP下INCLUDE和LIB文件夹分别复制到CodeBlocks的系统INCLUDE路径和系统LIB路径中，这两个文件夹分别包含了GSL函数库的头文件和库文件（以.a为扩展名）。
- ▶ 在程序源文件中包含相应的GSL头文件，在程序项目中添加连接库libgsl.a和libgslcblas.a，应用程序就能够调用GSL函数。

13.1.5 函数库配置举例



【例13.1】

使用LU分解法求解如下线性方程组。

$$\begin{bmatrix} 0.18 & 0.60 & 0.57 & 0.96 \\ 0.41 & 0.24 & 0.99 & 0.58 \\ 0.14 & 0.30 & 0.97 & 0.66 \\ 0.51 & 0.13 & 0.19 & 0.85 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \end{bmatrix}$$

13.1.5 函数库配置举例

例13.1

```
1 #include <stdio.h>
2 #include <gsl\gsl_linalg.h> //GSL线性代数函数
3 int main()
4 {
5     int s;
6     double A[]={0.18,0.60,0.57,0.96,0.41,0.24,0.99,0.58,0.14,
0.30,0.97,0.66,0.51,0.13,0.19,0.85};
7     double B[]={1.0,2.0,3.0,4.0};
8     gsl_matrix_view m;
9     gsl_vector_view b;
10    gsl_vector *x;
11    gsl_permutation *p;
12    m=gsl_matrix_view_array(A, 4, 4); //关联矩阵视图
13    b=gsl_vector_view_array(B, 4); //关联向量视图
14    x=gsl_vector_alloc(4); //建立求解向量空间
```

13.1.5 函数库配置举例

例13.1

```
15  p=gsl_permutation_alloc(4); //分配临时空间
16  gsl_linalg_LU_decomp(&m.matrix, p, &s); //LU分解
17  gsl_linalg_LU_solve(&m.matrix,p,&b.vector,x); //方程求解
18  printf("x=\n");
19  gsl_vector_fprintf(stdout,x, "%1f"); //向量输出
20  gsl_permutation_free(p); //释放临时空间
21  gsl_vector_free(x); //分配向量空间
22  return 0;
23 }
```

13.1.5 函数库配置举例

例13.1

程序运行屏幕



13.1.5 函数库配置举例

- ▶ 2. 在Visual C++环境下使用GSL
- ▶ 在VC使用GSL之前，需要将GSL的GCC库（.a）转换为VC库文件（.lib），方法是使用VC的“lib.exe”库管理命令行工具，命令形式为：

```
>lib /machine:i386 /def:libgsl.def  
>lib /machine:i386 /def:libgslcblas.def
```

- ▶ lib命令会在TEMP\LIB文件夹中产生libgsl.lib和libgslcblas.lib库文件。

13.1.5 函数库配置举例

- ▶ 使用前面的“添加路径法”，将TEMP\INCLUDE文件夹添加到VC系统INCLUDE路径中，将TEMP\LIB文件夹添加到VC系统LIB路径中。
- ▶ 在程序源文件中包含相应的GSL头文件和连接库预处理命令，就能调用GSL函数。

13.1.5 函数库配置举例



【例13.2】

求贝塞尔函数在5.0的值。

13.1.5 函数库配置举例

例13.2

```
1 #include <stdio.h>
2 #include <gsl\gsl_sf_bessel.h> //GSL特殊函数-贝塞尔函数
3 #pragma comment(lib,"libgsl.lib") //连接GSL函数库
4 #pragma comment(lib,"libgslcblas.lib")//连接GSL基础线性代数库
5 int main()
6 {
7     double x=5.0 , y;
8     y=gsl_sf_bessel_J0(x); //贝塞尔函数
9     printf("J0(5.0)=%.18f\n", y);
10    return 0;
11 }
```

程序运行结果如下：

J0(5.0)=-0.177596771314338260

CP 程序设计