



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计

Programming in C++



1011018

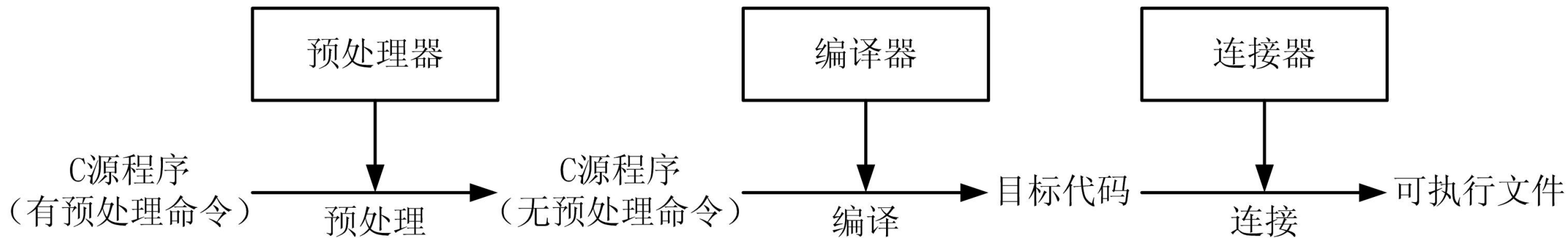
主讲：魏英，计算机学院

预处理命令

- ◆ 1、什么是预处理命令
- ◆ 2、宏定义

23.1 什么是预处理命令

图23.1 编译、连接处理过程



- ▶ 预处理命令不是C++本身的组成部分，更不是C++语句，它是C++标准规定的可以出现在C++源程序文件中的命令。
- ▶ 这些命令必须以“#”开头，结尾不加分号，可以放置在源程序中的任何位置，其有效范围是从出现位置开始到源程序文件末尾。

- ▶ 在C++源程序中允许用一个标识符来代表一个字符文本，称为宏，标识符为宏名。
- ▶ 宏是由宏定义命令事先定义的。预处理时，对程序中所有后续的宏名实例（称为宏引用），预处理器都用字符文本去替换，称为宏替换或宏展开。
- ▶ 宏定义通常用于定义程序中的符号常量、类型别名、运算式代换、语句代换等，其命令为`#define`，分为不带参数的宏定义和带参数的宏定义。

- ▶ 1. 不带参数的宏定义
- ▶ 不带参数的宏定义的命令形式为：

```
#define 宏名 字符文本
```

```
#define PI 3.1415926  
L=2*PI*r;
```

```
#define M y*y+5*y  
S=3*M+4*M+5*M;
```

```
L=2*3.1415926*r;
```

```
S=3*y*y+5*y+4*y*y+5*y+5*y*y  
+5*y;
```

- ▶ 宏定义只是简单置换，不作语法检查，因此，宏串中的每字符都是有效字符；
- ▶ 下例多余字符均导致编译出错：
 - ▶ `#define PI "3.141592"`
 - ▶ `c=2.0*PI*r;` 宏展开结果：`c=2.0*"3.141592"*r;`
 - ▶
 - ▶ `#define PI 3.141592;`
 - ▶ `c=2.0*PI*r;` 宏展开结果：`c=2.0*3.141592;*r;`

23.2 宏定义

【例23.1】

```
1  #include <iostream>
2  using namespace std;
3  #define PI  3.1415926 //不带参数的宏定义
4  int main()
5  {
6      double r,L,S,SQ,V;
7      cin >> r; //输入半径
8      L=2*PI*r; //计算圆周长
9      S=PI*r*r; //计算圆面积
10     SQ=4.0*PI*r*r; //计算圆球表面积
11     V=4.0*PI*r*r*r/3.0; //计算圆球体积
12     cout<<"L="<<L<<" ,S="<<S<<" ,SQ="<<SQ<<" ,V="<<V<<endl;
13     return 0;
14 }
```

- ▶ 2. 带参数的宏定义
- ▶ 带参数的宏定义的命令形式为：

```
#define 宏名(参数表) 字符文本
```

- ▶ 带参数的宏的引用形式为：

```
宏名(引用参数表)
```


23.2 宏定义

- ▶ 例如有宏定义

```
#define max(a,b) (((a) > (b)) ? (a) : (b))
```

- ▶ 程序代码：

```
L=max(x-y,x+y); //max宏引用
```

- ▶ 预处理时宏替换为：

```
L=(((x-y) > (x+y)) ? (x-y) : (x+y))
```

23.2 宏定义

【例23.2】

```
1  #include <iostream>
2  using namespace std;
3  int M1(int y)
4  {
5      return((y)*(y));
6  }
7  #define M2(y) ((y)*(y))
8  int main()
9  {
10     int i,j;
11     for (i=1,j=1;i<=5;i++) cout<<M1(j++)<<" "; //函数调用处理
12     cout << endl;
13     for (i=1,j=1;i<=5;i++) cout<<M2(j++)<<" "; //宏引用处理
14     cout << endl;
15     return 0;
```

- 为了保证宏展开的结果符合设计本意，应在宏串或实参字串中加入必要的括号；

```
#define S(r)  3.141592*r*r
int main( )
{ float a=2.0, b=3.0, area;
  area=S((a+b));  //替换为3.141592*(a+b)*(a+b)
  cout<<"area="<<area<<endl;
  return 0;
}
```

► 或者：

```
#define S(r)  3.141592*(r)*(r)
int main( )
{ float a=2.0, b=3.0, area;
  area=S(a+b);  //替换为3.141592*(a+b)*(a+b)
  cout<<"area="<<area<<endl;
  return 0;
}
```

- ▶ 3. #和##预处理运算
- ▶ #运算符的作用是文本参数“字符串化”，即出现在宏定义字符文本中的#把跟在后面的参数转换成一个C++字符串常量。

```
#define PRINT_MSG1(x) printf(#x);  
#define PRINT_MSG2(x) printf(x);  
PRINT_MSG1>Hello World); //正确  
PRINT_MSG1("Hello World"); //正确  
PRINT_MSG2>Hello World); //错误  
PRINT_MSG2("Hello World"); //正确
```

- ▶ ##运算符的作用是将两个字符文本连接成一个字符文本，如果其中一个字符文本是宏定义的参数，连接会在参数替换后发生。

```
#define SET1(arg) A##arg=arg;  
#define SET2(arg) Aarg=arg;  
SET1(1); //宏替换为 A1=1;  
SET2(1); //宏替换为 Aarg=1;
```

CP 程序设计