

高级语言程序设计

主讲：王老师



尚德机构

学习是一种信仰

第九章 编译预处理命令



本章内容

- 宏定义命令
- 文件包含命令
- 条件编译命令



编译预处理命令

编译预处理是指对源程序进行编译之前，先对源程序中的各种预处理命令进行处理；然后再将处理结果和源程序一起进行编译，以获得目标代码。



第一节 宏定义命令

C语言包括两种宏定义命令，即无参宏定义命令和有参宏定义命令。

通常，宏定义用于定义程序中的符号常量、类型别名、运算式代换和语句代换等。



无参宏定义

无参宏定义的一般格式：

【格式】#define 宏名 字符序列

【说明】

- 1、“宏名”是标识符的一种，命名规则与标识符相同，通常采用大写字母，以便与变量区分；“字符序列”可以是常量、表达式各种符号等。“宏名”和“字符序列”之间使用空格符分隔。
- 2、宏定义不是C语言的语句，所以不能在行尾加分号。否则，宏展开时会将分号作为字符串的一个字符，用于替换宏名。
- 3、宏展开时，预处理程序仅按宏定义简单地替换宏名，而不做任何检查。
- 4、对于双引号括起来的字符串内的字符，即使与宏名重名，也不进行宏展开。
- 5、宏定义命令#define出现在函数的外部，宏名的作用域是从宏定义开始到本文件结束。通常，宏定义命令放在文件的开头。
- 6、宏定义时，可以引用已定义的宏名，宏展开是逐层替换的。



无参宏定义

使用宏定义的优点在于：

- 1、提高源程序的可读性。
- 2、提高源程序的可修改性，修改宏定义中的“字符序列”可以起到一改全改的作用。
- 3、避免源程序中重复书写字符串。



有参宏定义

【格式】#define 宏名(形参表) 带参数的字符序列

有参宏定义的调用和宏展开格式：

宏名(实参表)

宏展开的过程是：若程序中有带实参的宏，则按照#define指定的字符序列从左至右进行替换；若宏定义中包含形参，则用实参直接替换宏定义命令行中相应的形参，其中实参可以是常量、变量或表达式；若宏定义的字符序列中的字符是非形参字符则替换时保留不变。



有参宏定义

【说明】

- 1、定义有参宏时，宏名与左圆括号之间不能留有空格。否则，C编译程序将空格以后的所有字符均作为替代字符串，而将该宏视为无参宏。
- 2、有参的宏展开只是将实参作为字符串，简单地替换形参字符串，而不进行任何语法检查。



★带参数的宏定义

❖一般形式: `#define 宏名(参数表) 宏体`

❖功能: 进行字符串替换, 并进行参数替换

例 `#define S(a,b) a*b`

.....

`area=S(3,2);`

宏展开: `area=3*2;`

❖宏展开: 形参用实参换, 其它字符保留

❖宏体及各形参外一般应加括号 `()`

例 `#define POWER(x) x*x`

`x=4; y=6;`

`z=POWER(x+y);`

宏展开: `z=x+y*x+y;`

一般写成: `#define POWER(x) ((x)*(x))`

宏展开: `z=((x+y)*(x+y));`

不能加空格

`*r*r`

例 使用带参数的宏

```
#include <stdio.h>
#define PI 3.1415926
#define S(r) PI*r*r
void main()
{ float a,area;
  a=3.6; area=S(a);
  printf("r=%f\narea=%f\n",a,area);
}
```

运行:

r=3.600000

area=40.715038

❖带参宏定义与函数的区别

- 函数调用时，先求实参表达式的值，再带入形参。
宏只进行简单字符替换，不求值。
- 函数调用在程序运行时处理和分配临时内存单元。
宏展开在编译时进行，不分配内存单元，无值传递和返回值。
- 函数要定义形实参且类型一致，宏无类型，其参数无类型。
- 函数只有一个返回值，宏可以设法得到几个结果。

例 使用宏带回几个结果

```
#include <stdio.h>
```

```
#define PI 3.1415926
```

```
#define CIRCLE(R,L,S,V) L=2*PI*R;S=PI*R*R;  
                        V=4.0/3.0*PI*R*R*R
```

```
void main()
```

```
{ float r,l,s,v;
```

```
  scanf("%f",&r);
```

```
  CIRCLE(r,l,s,v);
```

```
  printf("r=%6.2f,l=%6.2f,s=%6.2f,v=%6.2f\n",r,l,s,v);
```

```
}
```

宏展开后:

```
void main()
```

```
{ float r,l,s,v;
```

```
  scanf("%f",&r);
```

```
  l=2*3.1415926*r;s=3.1415926*r*r;v=4.0/3.0*3.1415926*r*r*r;
```

```
  printf("r=%6.2f,l=%6.2f,s=%6.2f,v=%6.2f\n",r,l,s,v);}
```

运行:

3.5 ↵

r=3.50,l=21.99,s=38.48,v=179.59

- 宏展开使源程序变长，函数调用源程序不变长。
- 宏替换不占运行时间，只占编译时间。
函数调用占运行时间。

例 用宏定义和函数实现同样的功能

```
#define MAX(x,y) (x)>(y)?(x):(y)
.....
main()
{ int a,b,c,d,t;
  .....
  t=MAX(a+b,c+d);
  .....
}
宏展开: t=(a+b)>(c+d)?(a+b):(c+d);
```

```
int max(int x,int y)
{ return(x>y?x:y);
}
main()
{ int a,b,c,d,t;
  .....
  t=max(a+b,c+d);
  .....
}
```

例 用宏代表输出格式

```
#include <stdio.h>
#define PR printf
#define NL "\n"
#define D "%d"
#define D1 D NL
#define D2 D D NL
#define D3 D D D NL
#define D4 D D D D NL
#define S "%s"
```

```
void main()
{ int a,b,c,d;
  char string[]="CHINA";
  a=1;b=2;c=3;d=4;
  PR(D1,a);
  PR(D2,a,b);
  PR(D3,a,b,c);
  PR(D4,a,b,c,d);
  PR(S,string);
}
```

运行结果：

```
1
12
123
1234
CHINA
```

课堂练习

1、设有宏定义`#define X 10-4`，则`printf("%d\n" , 6*X);` 的输出结果是_____。



课堂练习

1、设有宏定义`#define X 10-4`，则`printf("%d\n" , 6*X);` 的输出结果是_____。

答案：56

解析：宏定义命令将源程序中出现的X都用常数10-4替换，即在执行输出函数`printf("%d\n" , 6*X);` 时，输出6*X的值，把X用10-4替换，即 $6*10-4=56$ ，故输出的结果是56。



课堂练习

2、设`#define X 5+9`

则执行语句`printf("%d",3*X);`后输出的结果是_____。



课堂练习

2、设`#define X 5+9`

则执行语句`printf("%d",3*X);`后输出的结果是_____。

答案：24

解析：因为`#define X 5+9`，所以 $3*X=3*5+9=24$ ，执行语句`printf("%d",3*X);`后输出的结果是24。



第二節 文件包含命令

文件包含是指一个源文件可以将另一个源文件的全部内容包含进来。文件包含命令可以将多个模块共用的数据（如符号常量和数据结构）或函数，集中到一个单独的文件中。

文件包含命令的格式：

【格式一】`#include "文件名"`

【格式二】`#include <文件名>`

上述两种格式的区别仅在于：

- 1、格式一，系统首先到当前源文件所在的目录查找被包含文件，如果没有找到，再到系统指定的“包含文件目录”去查找，必要时在文件名前加上所在的路径。
- 2、格式二，直接到系统指定的“包含文件目录”去查找。





第二节 文件包含命令

【说明】

- 1、编译预处理时，预处理程序将查找指定的被包含文件，并将其复制到#include命令出现的位置上。
- 2、文件头部的被包含文件称为“标题文件”或“头部文件”，常以“h”（head）作为扩展名，简称头文件。
- 3、一条文件包含命令，只能指定一个被包含文件。
- 4、由于库函数以及某些宏定义都存放在系统文件中，一般都存放在系统指定的路径下，所以将存放库函数的头文件包含在源程序时一般建议采用格式二，即#include<文件名>。
- 5、文件包含命令允许嵌套，即一个被包含的文件中可以再使用文件包含命令包含另一个文件，而在该文件中还可以再包含其他文件。
- 6、若被包含文件的内容发生变化，则应该对包含此文件的所有源文件重新进行编译。



“文件包含”处理

★功能：一个源文件

★一般形式：`#include`

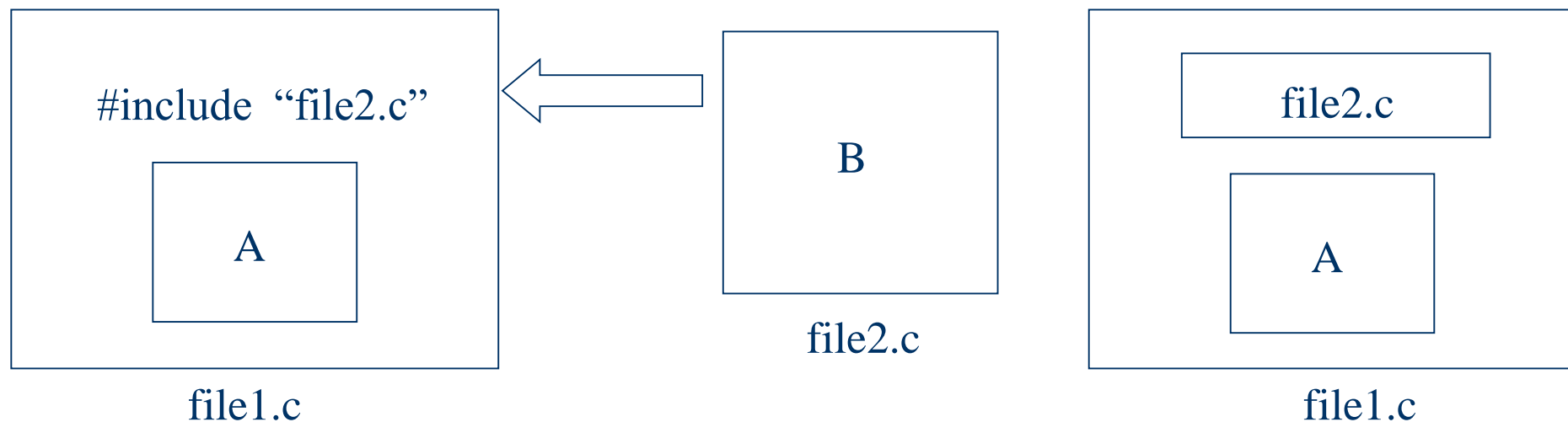
或 `#include <文件名>`

★处理过程：预编译时,用被包含文件的内容取代该预处理命令,再将“包含”后的文件作为一个源文件单位进行编译,得目标文件.obj

<> 直接按标准目录搜索

“ ” 先在当前目录搜索, 再搜索标准目录

可指定路径





课堂练习

3、使用系统函数sqrt()时，应在程序开头写上的编译预处理命令是（ ）

A. #include <stdio.h>

B. #include <string.h>

C. #include <math.h>

D. #include <stdlib.h>





课堂练习

3、使用系统函数sqrt()时，应在程序开头写上的编译预处理命令是 ()

A. #include <stdio.h>

B. #include <string.h>

C. #include <math.h>

D. #include <stdlib.h>

答案：C

解析：系统函数sqrt()是计算平方根的函数，则需要文件包含命令#include <math.h>。



课堂练习

4、下列属于文件包含的命令是 ()

A. #define N 25

B. #endif

C. #include "stdio.h"

D. #else





课堂练习

4、下列属于文件包含的命令是（ ）

A. #define N 25

B. #endif

C. #include "stdio.h"

D. #else

答案：C

解析：文件包含命令的格式：#include “文件名”、#include<文件名>。



第三节 条件编译命令

C语言提供了条件编译命令，使得用户可以选择对源程序的一部分内容进行编译，即不同的编译条件产生不同的目标程序。



条件编译命令

条件编译命令具有以下几种形式：

【格式一】`#ifdef` 标识符

程序段1

`[#else`

程序段2]

`#endif`

【功能】当“标识符”已经被`#define`命令定义，则编译程序段1；否则编译程序段2。其中
[]中的内容，即`#else`部分可以省略，即：

`#ifdef` 标识符

程序段1

`#endif`



条件编译命令

【格式二】 #ifndef 标识符

程序段1

[#else

程序段2]

#endif

【功能】当“标识符”没有被#define命令定义，则编译程序段1；否则编译程序段2。其中[]中的内容，即#else部分可以省略，即：

#ifndef 标识符

程序段1

#endif



条件编译命令

【格式三】 #if 表达式

程序段1

[#else

程序段2]

#endif

【功能】当表达式为“真”（非0）时，则编译程序段1；否则编译程序段2。其中，[]中的内容，即#else部分可以省略。

需要注意的是，其中的“表达式”是在编译阶段计算值的，所以此处的“表达式”不能是变量，必须是常量或用宏定义命令#define定义的标识符。





祝大家顺利通过考试!

