

Appendix A 预处理指令

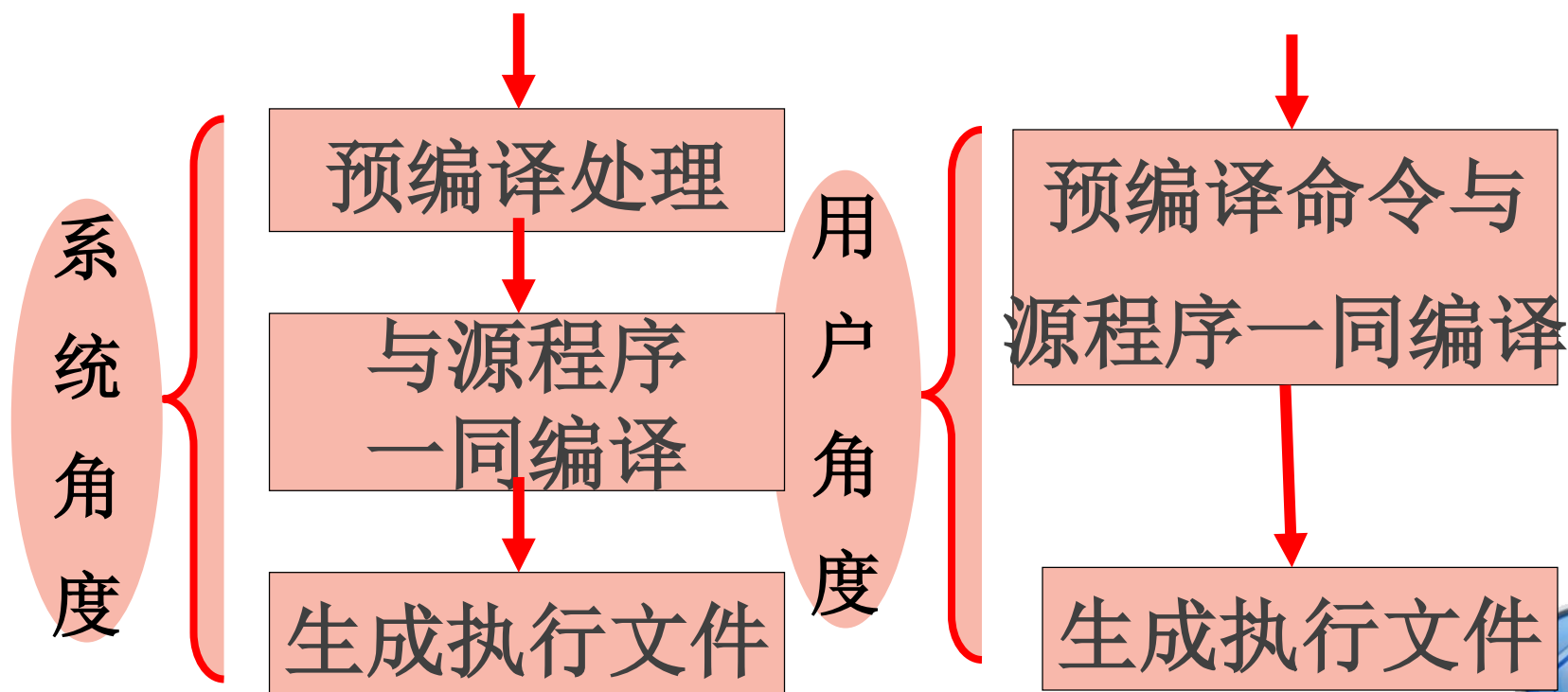
——preprocessing directives

A.1 宏

A.2 文件包含

A.3 条件编译

- 预处理命令:预处理命令不是**C**的组成部分，为了扩充**C**的功能**C**提供了预处理功能，预处理命令不能直接编译，要先编译，再和其源程序一起编译，执行过程如下



A.1 宏

A.1.1 不带参数的宏

➤ 格式: **#define** 标识符 字符串

◆ 例如: **#define PI 3.1415926**

◆ 作用: 指定用标识符**PI**来替代字符串
“**3.1415926**”

◆ 在编译时将宏名全部替换成字符串的过程称 “
宏展开”

◆ 可以用**#undef**命令终止宏定义的作用域

◆ 可引用已经定义的宏名, 可以层层置换



```
#define PI 3.1415926
```

预处理命令

```
int main( )
```

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

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

```
l=2.0*PI*r;
```

```
s=PI*r*r;
```

```
v=3.0/4*PI*r*r*r;
```

```
printf("l=%f\ns=%f\nv=%f",l,s,v);
```

```
return 0;
```

```
}
```

如果将#define PI中PI的值修改一次，程序中所有PI值都全部自动更改。

A.1 宏

A.1.2 带参数的宏function-like macro

➤ 格式 **#define** 宏名(参数表) 字符串

◆ 例如: **#define S(a, b) a*b**

◆ **area= S(3, 2)**

◆ 作用: 指定宏名、参数和字符串表达式, 置换时从左到右将实参(表达式)代替形参(保留非参数字符)。

A.2 文件包含

- “文件包含” 指在一个文件中可以包含另一个文件（内容）；这样可以减少重复工作(公用模块)。
- 一般形式：**#include "file1.c"**
或：**#include <stdio.h>**
- **stdio.h math.h string.h malloc.h
stdlib.h**

A.3 条件编译

- 一般情况下，编译一个源程序都是所有语句行都编译(前面所有例子)，为了需要可以将满足条件的一部分语句进行编译，称条件编译。
- 条件编译命令的**3**种形式：

<pre>#ifdef 标识符 程序段1</pre>	<pre>#ifndef 标识符 程序段1</pre>	<pre>#if 表达式 程序段1</pre>
<pre>#else 程序段2</pre>	<pre>#else 程序段2</pre>	<pre>#else 程序段2</pre>
<pre>#endif /*(1)*/</pre>	<pre>#endif /*(2)*/</pre>	<pre>#endif /*(3)*/</pre>

A.3 条件编译

➤ 格式(1)的说明:

#ifdef 标识符

程序段1

[#else

程序段2]



可以没有

#endif

➤ 当所指定的标识符被**#define**命令定义过，则在编译时编译程序段1，否则编译程序段2。

A.3 条件编译

- 格式(2)的说明: [与格式(1)相反]

#ifndef 标识符

程序段1

[#else

程序段2]



可以没有

#endif

- 当所指定的标识符未被**#define**命令定义, 则在编译时编译程序段1, 否则编译程序段2。

A.3 条件编译

- 格式**(3)**的说明:

#if 表达式

程序段**1**

[#else

程序段**2]**



可以没有

#endif

- 当表达式值为真(非**0**)时，则在编译时编译程序段**1**，否则编译程序段**2**。