

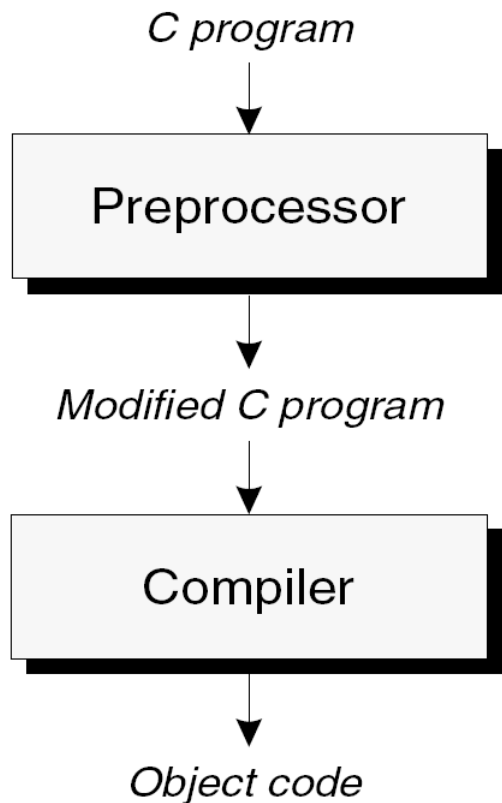


第14章

预处理器

预处理器的工作方式

- 在编译过程中，预处理器的角色如图所示：



预处理指令——3种类型

- 宏定义：
 - **#define** 宏 定义, eg **PI 3.14**
 - **#undef** 删除一个宏定义
- 文件包含：
 - **#include** 包含一个指定文件的内容到程序中
- 条件编译：
 - **#if**、**#ifdef**、**#ifndef**、**#elif**、**#else**和**#endif**
 - 根据测试条件将一段代码块包含到程序中或排除在程序之外。
 - eg, ppt 页面隐藏、选择翻译

第二章中的 celsius.c

```
/* Converts a Fahrenheit temperature to Celsius */  
  
#include <stdio.h>  
  
#define FREEZING_PT 32.0f  
#define SCALE_FACTOR (5.0f / 9.0f)  
  
int main(void)  
{  
    float fahrenheit, celsius;  
  
    printf("Enter Fahrenheit temperature: ");  
    scanf("%f", &fahrenheit);  
  
    celsius = (fahrenheit - FREEZING_PT) *  
    SCALE_FACTOR;  
    printf("Celsius equivalent is: %.1f\n", celsius);  
  
    return 0;  
}
```

预处理过后为

Blank line

注释删除

Blank line

Lines brought in from stdio.h

头文件加载

Blank line

Blank line

Blank line

宏替换/代入

Blank line

```
int main(void)
```

```
{
```

```
    float fahrenheit, celsius;
```

```
    printf("Enter Fahrenheit temperature: ");
```

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

```
    celsius = (fahrenheit - 32.0f) * (5.0f / 9.0f);
```

```
    printf("Celsius equivalent is: %.1f\n", celsius);
```

```
    return 0;
```

```
}
```

宏替换/代入

预处理指令由预处理器处理
编辑器看不到

预处理指令

- 指令可以出现在程序中任何地方
 - 通常 `#define` 和 `#include` 指令放在文件的开始
 - 其他指令放在后面，甚至在函数定义的中间

宏定义

- 简单宏
- 带参数的宏

带参数的宏

- 格式:

- `#define` 标识符 (`x1, x2, ..., xn`)
替换列表。eg,

- `#define average(x,y) (x+y)/2`

- 注意: 宏名和左括号间不能有空格

- 预处理器遇到空格就认为宏名结束, 就变成简单宏

- `define f(x1,x2) x1+x2`

- `define f (x1,x2) x1+x2`

带参数的宏

- 带参数的宏举例：

```
#define MAX(x,y) ((x)>(y)?(x):(y))
```

```
#define IS_EVEN(n) ((n)%2==0)
```

- 调用这些宏：

```
i = MAX(j+k, m-n);
```

```
if (IS_EVEN(i)) i++;
```

- 宏替换后：

```
i = ((j+k)>(m-n)?(j+k):(m-n));
```

```
if (((i)%2==0)) i++;
```

宏定义中的圆括号

- 带参数宏，替换列表中参数圆括号是必要的
 - `#define SCALE(x) (x*10)`
- 在预处理过程中，语句
 - `j = SCALE(i+1);`
- 变为
 - `j = (i+1*10);`

条件编译

- 使预处理器根据条件测试结果来决定是否包含程序中的一些片断
 - eg. 选择翻译
- 惯用于调试程序
 - `printf()` 测试或跟踪关键变量的值——调试代码
 - 调试完后，不需运行调试代码，但需保留以备后用
- 保留调试代码，但让编译器忽略它们。
 - 隐藏ppt

#if指令和#endif指令

- 设置调试开关，定义一个宏
 - `#define DEBUG 1`
 - 真(1)为调试模式：保留（编译）调试代码
 - 假(0)为非调试模式：删除（不编译）调试代码
- 然后，将调试代码用`#if`和`#endif`框住：
 - `#if DEBUG`
 - `printf("Value of i: %d\n", i);`
 - `printf("Value of j: %d\n", j);`
 - `#endif`
- 调试结束后，
 - `#define DEBUG 0`
- 预处理器则会将这4行代码都删除。

测试宏真假

#if 与 defined 结合

- 不必给DEBUG值:

```
#define DEBUG
```

- 例子:

```
#define DEBUG
```

```
.....
```

```
#if defined(DEBUG)
```

测试宏有无

```
...
```

```
#endif
```

- `defined(DEBUG)` 返回true
- 可省略括号:

```
#if defined DEBUG
```

#ifdef指令和#ifndef指令

- **#ifdef**: 测试标识符是否已经定义为宏:

```
#ifdef identifier  
    statement;
```

```
#endif
```

- **#ifndef**: 测试标识符是否还未被定义为宏:

```
#ifndef identifier  
    statement;
```

```
#endif
```

```
#include <stdio.h>
#define FACTOR 0.9f
#define DEBUG
int main(void){
    int height;
    float weight, stdwt;
    char sex;
    puts("输入性别, 男性用m表示, 其它字符表女性.");
    scanf("%c", &sex);
    puts("输入身高(cm).");
    scanf("%d", &height);
    #ifdef DEBUG
    printf("%c\t%d\n",sex,height);
    #endif
    if (sex == 'm')
        stdwt = (height - 100) * FACTOR;
    else
        stdwt = (height - 100) * FACTOR - 2.5;
    printf("你的标准体重应是%.1fkg\n.", stdwt);
    return 0;
}
```


#elif指令和#else指令

- **#elif**和**#else**与**#if**、**#ifdef**和**#ifndef**组合，来测试一系列条件：
- **#if expr1**
 - 当表达式 1非 0时需要包含的代码
- **#elif expr2**
 - 当表达式 1为 0但表达式 2非 0时需要包含的代码
- **#else**
 - 其他情况下需要包含的代码
- **#endif**

使用条件编译

- 编写在多台机器或多种操作系统之间可移植的程序。
- 例如：
 - `#if defined(WIN32)`
 - ...
 - `#elif defined(MAC_OS)`
 - ...
 - `#elif defined(LINUX)`
 - ...
 - `#endif`