通过conditional命令可以让预处理器选择代码是否需要加入到最终的token中进行编译。预处理器条件指令可以进行算术判断，宏定义或者通过defined进行判断。

预处理的条件语句类似C语言的if语句，但是这两者有所不同。if语句是程序执行时进行判断的，目的是程序执行时，可以动态改变执行的代码块。预处理中的条件语句是在编译期间判断代码是否需要编译，目的是在编译时判断需要那部分代码。

不过现在这个区别逐渐模糊。现代编译器常常编译时会对if判断。如果你知道你所用的编译器可以这么做，你可以通过if判断常量，来让代码更加具有可读性。

你只能通过conditional指令来包含或者不包含代码，而不能做其他用，比如type definition或者其他预处理指令，而且要保证语法上的合法性。

GCC version 3开始对这些不执行的代码进行优化，即便没有显示的优化，老一些的编译器在优化时会这么做。【需要在v4.9上测试，在4.4.3上通过-O3 –E没有体现出来】

4.1 Conditonal Uses

有三个理由使用条件判断。

1，程序依赖不同的机器或者操作系统。

2，通过同一个源代码编译不同的程序。

3，一个总是fail的语句，但是为了给以后的维护带来参考。

不需要系统特定逻辑或者不需要调试信息的简单程序，通常不需要预处理条件判断。

4.2 Conditional Syntax

C预处理器的条件指令，以#if, #ifdef 或者#ifndef开始。

4.2.1 Ifdef

```
#ifdef Macro
.....controlled text
#endif
```

这个代码块称之为conditional gruop。controlled text当且仅当Macro定义时，预处理器会输出。在Macro定义时，条件判断为真，未定义时，条件为假。【译注：如果定义了，不过值为0，则#ifdef macro依然为真】

在controller test中可以有其他的预处理指令。这些指令在条件判断为真是执行。在conditional group中可以嵌套conditonal group，不过必须完全嵌套，一个#endif对应最近的一个#ifdef（或者#if, #ifndef）。conditonal group也不能跨文件。

controlled text，即便判断为假，也会执行initial transformation和tokenization，因此controlled text必须是合法的C语句。

4.2.2 If

```
#if expression
...controlled text
```

#endif

expression是C语句中整形的判断语句。可能是：

1，整数常量

2，字符常量

3，算术运算，加减乘除，位运算，逻辑运算

4，宏，在计算expression的值以前进行宏扩展

5，defined操作符。可以在#if expression中判断一个宏是否定义。

6，所有不是宏的标示符，都会认为是0。这样可以用#if Macor来代替#ifdef Macro，当macro定义时，Macro总是个非零值【译注，可以通过-D，-U来修改】。function-like的宏，如果不含括号，同样也会认为是0。

-Wundef可以对#if语句使用不是宏的标示符的情况发出警告。

预处理器不对类型判断。所以sizeof，enum等所有涉及类型的常量或者运算，在if语句中均不可用。这些标示符会被扩展为0或者报错。

预处理器计算expression的值。会将整数按照编译器定义的最宽的类型来计算，在大多数支持GCC的机器上是64bit。这个和编译器计算常量表达式不同，有时候，两者计算的值还不一致。只要计算的值非零，那么#if就判断为真，其他的值就是假。

4.2.3 Defined

defined在#if, #elif中判断一个标示符是不是一个宏。defined name和defined (name)，如果在判断时定位为一个宏，那么就为真，否则为假。这样#if defined Macro 和#ifdef Macro一样。

【注：#if判断的是expression的值是否为真，需要进行计算，#ifdef 是判断一个宏是否定义】

【注：Sample】

```
#define test 0
#if test
        printf("test if directive\n");
#endif

#if defined test
        printf("test if defined directive\n");
#endif

#ifdef test
        printf("test ifdef directive\n");
#endif
------------------------------------------------------------
test if defined directive
test ifdef directive
```

【注：sample end】

defined在需要判断超过一个宏的时候比较有用。比如：

```
#if defined (__var__) || defined (__ns1600__)
#if defined BUFSIZE && BUFSIZE >= 1024
```

如果defined是通过宏扩展而来的，那么C标准规定这种情况是未定义的。CPP可以处理这种情况。不过你可以通过

制定-pedantic来限制这种情况。

4.2.4 Else

给if判断增加一个分支

#if expression

text-if-true

#else

text-if-false

#endif

同样的，也可以给#ifdef, #ifndef使用else。

4.2.5 Elif

#if expression

...

#elif expression

....

#elif expression

...

#endif

#if expression

...

#else

#if expression

....

#else

#if expression

...

#endif

4.3 删除代码

当想删除一部分代码的功能，但是想保留这些代码时，可以用#if 0来注释。

不要用#if 0来注释不合法的C代码，要使用真正的注释符号。#if 0中的代码也会进行符号化。