宏

编译预处理指令

- #开头的是编译预处理指令
- 它们不是C语言的成分，但是C语言程序离不开它们
- #define用来定义一个宏

#define

- #define <名字> <值>
- 注意没有结尾的分号，因为不是C的语句
- 名字必须是一个单词，值可以是各种东西
- 在C语言的编译器开始编译之前，编译预处理程序（cpp）会把程序中的名字换成值
  - 完全的文本替换
- gcc —save-temps

宏

- 如果一个宏的值中有其他的宏的名字，也是会被替换的
- 如果一个宏的值超过一行，最后一行之前的行末需要加\
- 宏的值后面出现的注释不会被当作宏的值的一部分

没有值的宏

- #define _DEBUG
- 这类宏是用于条件编译的，后面有其他的编译预处理指令来检查这个宏是否已经被定义过了

预定义的宏

- __LINE__
- __FILE__
- __DATE__
- __TIME__
- __STDC__

带参数的宏

像函数的宏

- #define cube(x) ((x)*(x)*(x))
- 宏可以带参数

错误定义的宏

- #define RADTODEG(x) (x * 57.29578)
- #define RADTODEG(x) (x) * 57.29578

带参数的宏的原则

- 一切都要括号
  - 整个值要括号
  - 参数出现的每个地方都要括号
- #define RADTODEG(x) ((x) * 57.29578)

带参数的宏

- 可以带多个参数
  - #define MIN(a,b) ((a)>(b)?(b):(a))
- 也可以组合（嵌套）使用其他宏

分号?

```
#define PRETTY_PRINT(msg) printf(msg);


if (n < 10)
    PRETTY_PRINT("n is less than 10");
else
    PRETTY_PRINT("n is at least 10");
```

带参数的宏

- 在大型程序的代码中使用非常普遍
- 可以非常复杂，如“产生”函数
  - 在#和##这两个运算符的帮助下
- 存在中西方文化差异
- 部分宏会被inline函数替代

其他编译预处理指令

- 条件编译
- error
- …