一、C语言编译过程

C语言的编译过程：

预处理、编译、汇编、链接

gcc -E hello.c -o hello.i　　1、预处理
gcc -S hello.i –o hello.s　　2、编译
gcc -c hello.s -o hello.o　　3、汇编
gcc hello.o -o hello_elf　　4、链接

1：预编译

将.c 中的头文件展开、宏展开

生成的文件是.i文件

2：编译

将预处理之后的.i 文件生成 .s 汇编文件

3、汇编

将.s汇编文件生成.o 目标文件

4、链接

将.o 文件链接成目标文件

二、include

#include<>//用尖括号包含头文件，在系统指定的路径下找头文件

#include "" //用双引号包含头文件，先在当前目录下找头文件，找不到，再到系统指定的路径下找。

注意：include 经常用来包含头文件，可以包含 .c 文件，但是大家不要包含.c

因为include包含的文件会在预编译被展开，如果一个.c 被包含多次，展开多次，会导致函数重复定义。
所以不要包含.c 文件。

注意：预处理只是对include 等预处理操作进行处理并不会进行语法检查
这个阶段有语法错误也不会报错，第二个阶段即编译阶段才进行语法检查。

三、define

定义宏用define 去定义
宏是在预编译的时候进行替换。

1、不带参宏

#define PI 3.14

在预编译的时候如果代码中出现了PI 就用 3.14去替换。

宏的好处：只要修改宏定义，其他地方在预编译的时候就会重新替换。

注意：宏定义后边不要加分号。

```c
#include <stdio.h>

//宏定义的好处是只要改变了定义是的常量表达式，则代码中只要使用这个宏定义的位置都会改变
#define PI 3.1415926

int main(int argc, char *argv[])
{
 printf("PI = %lf\n", PI);

 double d = PI;
 printf("d = %lf\n", d);

 return 0;
}
```

执行结果

```
Starting C:\Users\lzx\De
\debug\01_define.exe...
PI = 3.141593
d = 3.141593
C:\Users\lzx\Desktop\src
\01_define.exe exited wi
```

宏定义的作用范围，从定义的地方到本文件末尾。

如果想在中间终止宏的定义范围

#undef PI //终止PI的作用

2、带参宏

#define S(a,b) a*b

注意带参宏的形参 a和b没有类型名，

S(2,4) 将来在预处理的时候替换成 实参替代字符串的形参，其他字符保留，2 * 4

```c
#include <stdio.h>

//带参宏
//带参宏类似于一个简单的函数，将函数的参数进行设置，就可以传递给对应的表达式
//#define S(a, b) a*b
#define S(a, b) ((a)*(b))

int main(int argc, char *argv[])
{
  printf("%d\n", S(2, 4));
  //注意：宏定义只是简单的替换，不会自动加括号
  //带参宏1：2 + 8 * 4 = 34
  //带参宏2：((2 + 8) * (4)) = 40
  printf("%d\n", S(2 + 8, 4));

  return 0;
}
```

3、带参宏和带参函数的区别

带参宏被调用多少次就会展开多少次，执行代码的时候没有函数调用的过程，不需要压栈弹栈。所以带参宏，是浪费了空间，因为被展开多次，节省时间。

带参函数，代码只有一份，存在代码段，调用的时候去代码段取指令，调用的时候要压栈弹栈。有个调用的过程。所以说，带参函数是浪费了时间，节省了空间。

带参函数的形参是有类型的，带参宏的形参没有类型名。

如果功能实现的代码相对简单，并且不需要开辟太多的空间，可以选择使用带参宏，但是大多数情况都会使用函数

四、选择性编译

1、

#ifdef AAA

    代码段一

#else

    代码段二

#endif

如果在当前.c ifdef 上边定义过AAA ，就编译代码段一，否则编译代码段二

注意和if else语句的区别，if else 语句都会被编译，通过条件选择性执行代码

而 选择性编译，只有一块代码被编译

```c
#define  AAA

int main(int argc, char *argv[])
{
 #ifdef  AAA
 printf("hello kitty!!\n");
 #else
 printf("hello 千锋edu\n");
 #endif
  return 0;
}
```

2、

#ifndef AAA

    代码段一

#else

    代码段二

#endif

和第一种互补。

这种方法，经常用在防止头文件重复包含。

常用于多文件编程中.h的第一行就是#ifndef，最后一行就是#endif

3、

#if 表达式

程序段一

#else

程序段二

#endif

如果表达式为真，编译第一段代码，否则编译第二段代码

这种形式一般用于注释多行代码

#if 0

...

#endif

选择性编译都是在预编译阶段干的事情。