

C基础：手工编译链接

在Windows平台上，C的编译器一般为 `cl.exe`，连接器为 `link.exe`，汇编器为 `ml.exe`

C源程序一般经过以下阶段变为可执行程序：

1. 预处理
2. 编译
3. 汇编
4. 链接

在Linux环境下，使用gcc编译器以上四步分别为：

```
gcc -E hello.c -o hello.i
gcc -S hello.i -o hello.s
gcc -c hello.s -o hello.o
gcc hello.o -o hello
```

一般来说，

1. 预处理阶段，进行文件包含，宏替换等
2. 编译阶段，检查语法错误或警告，在无错误后将代码翻译成汇编代码
3. 汇编阶段，将编译阶段生成的汇编代码再加工，生成二进制目标文件
4. 链接阶段，将目标文件与相关库或者其他依赖进行链接，最终生成可执行文件

若使用Windows手工编译则可以进行如下操作：

```
cl /c hello.c
link hello.obj
```


C基础：头文件包含

在C语言中，可以用 `#include <xxx.h>` 或 `#include "xxx.h"` 预处理指令进行头文件包含，两者区别如下：

1. `<>` 表示先去系统目录中找头文件，如果没有在到当前目录下找
2. `" "` 表示首先在当前目录下寻找，如果找不到，再到系统目录中寻找

C基础：入口函数

一般，程序员认为C的入口函数为**main**函数，其实不然。真正的入口函数和运行平台有关，但都属于**CRT**中的启动函数。

名称	语言
 test.exe!main() 行 15	C
test.exe!invoke_main() 行 78	C++
test.exe!__scrt_common_main_seh() 行 288	C++
test.exe!__scrt_common_main() 行 331	C++
test.exe!mainCRTStartup() 行 17	C++
kernel32.dll!@BaseThreadInitThunk@12()	未...
ntdll.dll!_RtlUserThreadStart()	未...
ntdll.dll!_RtlUserThreadStart@8()	未...

在VS2017中，可以看到main函数被其他函数调用了，而这些函数就是windows平台上相关的C运行时函数，在main函数启动前，初始化线程、用户堆、命令行、环境变量等等一系列的系统资源。然后将程序控制权交给程序员手上