

C & C++ 程序设计

C & C++ 简介

张晓平

数学与统计学院

1. 编译 C 程序的工作原理

编译 C 程序的工作原理

编译 C 程序的工作原理

C 是一种高级语言，它需要编译器将其转换为可执行代码，以使得程序能在机器上运行。

以下介绍在 MAC 或 Linux 上使用 gcc 编译器的几个步骤。

- (1) 首先使用编辑器（如 vi 或 emacs 等）创建一个 C 程序，并将其保存为 hello.c。

```
$ emacs hello.c
```

编译 C 程序的工作原理

在编辑界面输入以下内容：

```
/*  
 *   A simple c code  
 */  
#include <stdio.h>  
int main(void)  
{  
    printf("Hello World!\n");  
    return 0;  
}
```

(2) 然后用以下命令编译，并查看当前目录下的文件。

```
$ gcc -Wall hello.c -o hello
$ ls
hello    hello.c
```

- 选项 `-Wall` 启动所有编译器的警告信息。建议使用该选项以生成更好的代码。
- 选项 `-o` 用来制定输出文件名。如果缺省该选项，则输出文件将默认为 `a.out`。

(3) 编译通过后，将会生成可执行文件。可用以下命令来运行：

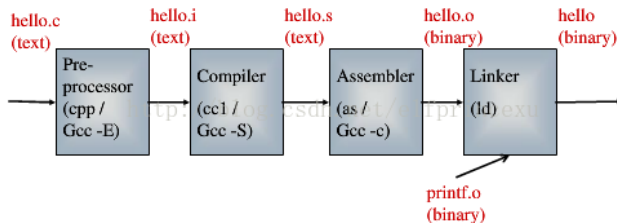
```
$ ./hello  
Hello World!
```


编译 C 程序的工作原理

编译器将一个 C 程序转换为一个可执行文件，需经历了 4 个阶段：

- 预处理
- 编译
- 汇编
- 链接

编译工具链的基本工作流程



一个“hello world”程序的演变历程

编译 C 程序的工作原理

执行以下命令，会在当前目录下生成所有的中间文件以及可执行文件。

```
$ gcc -Wall -save-temps hello.c -o hello
$ ls
hello      hello.c    hello.o
hello.i    hello.s
```

(1) 预处理阶段

```
$ gcc -E hello.c -o hello.i
```

在该阶段，

- 去掉注释
- 宏的展开
- 头文件的展开
- 选项 `-E` 的作用是让 `gcc` 在预处理结束后停止编译过程。

编译 C 程序的工作原理

```
$ less hello.i
$ ls
...
extern int printf (const char *__restrict __format, ...)
;
...
# 868 "/usr/include/stdio.h" 3 4
# 5 "hello.c" 2
# 5 "hello.c"

int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

(2) 编译阶段

```
$ gcc -S hello.i -o hello.s
```

在该阶段，

- 检查代码的规范性、是否有语法错误等
- 检查无误后，将代码翻译成汇编语言
- 选项 `-S` 只进行编译，不进行汇编，生成汇编代码。

编译 C 程序的工作原理

```
$ less hello.s
...
movl    $0, -4(%rbp)
movl    $5, -8(%rbp)
movl    $4, -12(%rbp)
movl    -8(%rbp), %eax
addl    -12(%rbp), %eax
movl    %eax, %esi
movb    $0, %al
callq   _printf
xorl    %esi, %esi
...
```

(3) 汇编阶段

```
$ gcc -c hello.s -o hello.o
```

在该阶段，

- 将通过汇编器将 `hello.s` 转换成二进制机器指令文件 `hello.o`。
- 只会将现有代码转换成机器语言，而诸如 `printf()` 的函数调用则不会。
- 选项 `-c` 的作用是将汇编代码转换为二进制目标代码。

编译 C 程序的工作原理

```
$ less hello.o
<CF><FA><ED><FE>^G^@^@^A^C^@^@^@^A^@^@^@^D^@
^@^@^@^B^@^@^@ ^@^@^@^@^@^@^Y^@^@^@<88>^A^@^
^@^@^@^@^@^@
...
```


(4) 链接阶段

这是最后一个阶段，将完成所有函数调用及其定义的链接工作。链接器知道所有这些函数在何处执行。链接器也会做一些额外的工作，以添加一些启动和结束程序所需的额外代码。

编译 C 程序的工作原理

在命令行中输入以下命令，可看出从目标文件到可执行文件时文件大小的变化。这是因为链接器为我们的程序添加了额外的代码。

```
$ size hello.o
__TEXT    __DATA    __OBJC    others      dec          hex
145        0         0         32         177          b1
$ size hello
__TEXT    __DATA    __OBJC    others      dec          hex
4096      4096      0  4294971392  4294979584  100003000
```