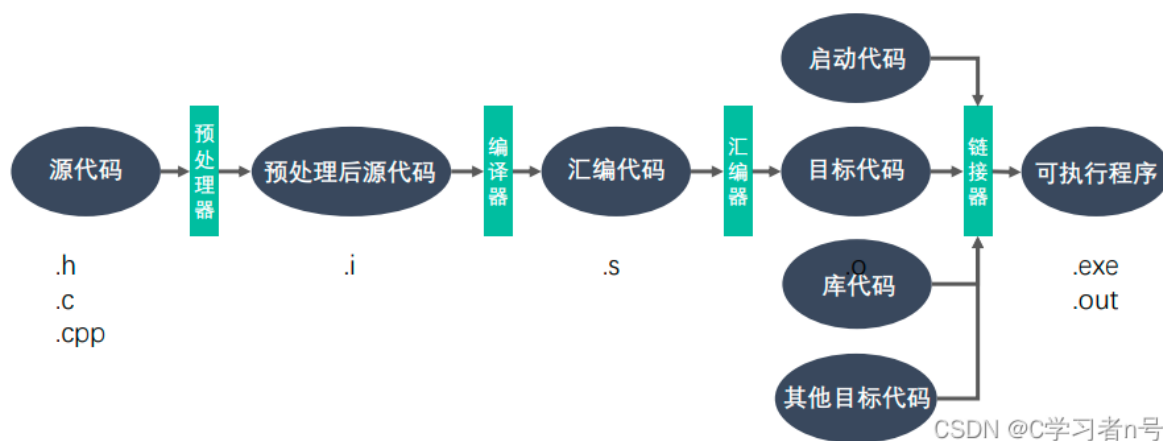


GCC介绍

1. GCC原名GUN C语言编译器（GNU C Compiler）
2. GCC（GNU Compiler Collection，GNU编译器套件）是由 GNU 开发的编程语言译器。GNU 编译器套件包括 C、C++、Objective-C、Java、Ada 和 Go 语言前端，也包括了这些语言的库（如 libstdc++，libgcj等）
3. GCC不仅支持C的许多方言，也可以区别不同的C语言标准；可以使用命令行选项来控制编译器在翻译源代码时应该遵循哪个 C 标准。例如，当使用命令行参数 `-std=c99` 启动 GCC 时，编译器支持 C99 标准。
4. 安装命令 `sudo apt install (gcc / g++)`
5. 查看版本 `gcc / g++ -v/--version`
6. gcc和g++是GCC编译套件中的一种。

gcc和g++的工作流程



1. 预处理阶段

在这个阶段会把头文件展开、宏替换、注释去掉，并生成一个.i文件

```
gcc -E test.c -o test.i
```

```
#include <stdio.h>

//这是一个自定义宏
#define MAX 10000

int main()
{
    //这是一个局部变量
    int i = MAX;
    printf("%d\n", i);

    return 0;
}
```

2. 编译阶段

这个阶段生成汇编语言代码，文件名带有.s

```
gcc -S test.i -o test.s
```

3. 汇编阶段

该阶段将编译出来的汇编代码通过汇编器生成二进制的代码，文件名后缀.o

```
gcc -c test.s -o test.o
```

4. 链接阶段

该阶段通过连接器，链接到其他需要的代码，并且和当前程序链接，生成可执行文件。

```
gcc -o test test.o
```

5. 可以一步到位生成可执行程序

```
gcc -o test test.c
```

gcc和g++的相关参数

- -E -S -c -o

这四条参数是在预处理、编译、汇编和链接的时候所需要的。

- -Wall -w

第一个是生成所有警告信息，四个阶段中只有预处理阶段不会报警告信息。

比如定义一个变量，但是不对它进行任何的使用，就会有警告信息。

第二个表示不生成任何警告信息。

- -D

在程序编译的时候指定一个宏。

```
#include <stdio.h>

//这是一个自定义宏
#define MAX 10000

int main()
{
#ifdef MyDef
printf("宏信息\n");
#endif

//这是一个局部变量
int i = MAX;
int j = 0;
printf("%d\n", i);

return 0;
}
```

```
gcc -o test test.c -DMyDef
```

在编译的时候指定了MyDef宏，那么在代码中的宏信息输出就会被执行。反之，不会执行这个参数。

- -std
指定方言，也就是c的版本问题。有的代码在一些版本中是无法通过的。

```
for (int loop = 0; loop < 3; loop++)  
{  
    printf("%d\n", loop);  
}
```

```
gcc -o test test.c -std=c89
```

如上，在c89中，c的循环变量不能定义在for里面，所以在编译的时候会报错。让我们尝试用其他c的方言。

```
test.c: In function 'main':  
test.c:13:5: error: 'for' loop initial declarations are only allowed in C99 or  
C11 mode  
   13 |     for (int loop = 0; loop < 3; loop++)  
      |           ^~~~  
test.c:13:5: note: use option '-std=c99', '-std=gnu99', '-std=c11' or '-  
std=gnu11' to compile your code
```

- -l -L
-l --> 指定程序编译需要的库名
-L --> 指定程序编译时依赖的库的路径
在生成动态库和静态库的时候需要使用。
-l的一个应用就是利用gcc编译c++程序，gcc -o test test.cpp -lstdc++
- -I(大写的i)
指定 include 包含文件的搜索目录
- -g
在编译的时候，生成调试信息，该程序可以被调试器调试
- -shared
生成共享目标文件，通常用在建立动态库时使用。

gcc和g++的误区（区别）

误区一： gcc 只能编译 c 代码， g++ 只能编译 c++ 代码

- 后缀为 .c 的， gcc 把它当作是 C 程序， 而 g++ 当作是 c++ 程序
- 后缀为 .cpp 的， 两者都会认为是 C++ 程序， C++ 的语法规则更加严谨一些
- 编译阶段， g++ 会调用 gcc， 对于 C++ 代码， 两者是等价的， 但是因为 gcc 命令不能自动和 C++ 程序使用的库链接， 所以通常用 g++ 来完成链接， 为了统一起见， 干脆编译/链接统统用 g++ 了， 这就给人一种错觉， 好像 cpp 程序只能用 g++ 似的

误区二： 编译只能用 gcc， 链接只能用 g++

- 严格来说， 这句话不算错误， 但是它混淆了概念， 应该这样说： 编译可以用 gcc/g++， 而链接可以用 g++ 或者 gcc -lstdc++。
- gcc 命令不能自动和C++程序使用的库链接， 所以通常使用 g++ 来完成链接。 但在编译阶段， g++ 会自动调用 gcc， 二者等价

