GCC -o选项：指定输出文件

< 上一页 下一页 >

gcc -o 选项用来指定输出文件，如果不使用 -o 选项，那么将采用默认的输出文件。例如默认情况下，生成的可执行文件的名字默认为 a.out。

如下是 gcc -o 指令的使用语法格式：

```
[root@bogon demo]# gcc [-E|-S|-c] [infile] [-o outfile]
```

> 其中，用方括号 [] 括起来的部分可以忽略。

[infile] 表示输入文件（也即要处理的文件），它可以是源文件、汇编文件或者目标文件；[outfile] 表示输出文件（也即处理的结果），可以是预处理文件、目标文件、可执行文件等。

值得一提的是，通常情况下 [infile] 处放置一个文件，但根据实际需要也可以放置多个文件，表示有多个输入文件（后续会给出实例）。

GCC -o选项使用举例

1) 将源文件作为输入文件，将可执行文件作为输出文件，也即完整地编译整个程序：

```
$ gcc main.c func.c -o app.out
```

将 main.c 和 func.c 两个源文件编译成一个可执行文件，其名字为 app.out。如果不使用 -o 选项，那么将生成名字为 a.out 的可执行文件。

2) 将源文件作为输入文件，将目标文件作为输出文件，也即只编译不链接：

```
$ gcc -c main.c -o a.o
```

将源文件 main.c 编译为目标文件 a.o。如果不使用 -o 选项，那么将生成名为 main.o 的目标文件。

3) 将源文件作为输入文件，将预处理文件作为输出文件，也即只进行预处理操作：

```
$ gcc -E main.c -o demo.i
```

对源文件 main.c 进行预处理操作，并将结果放在 demo.i 文件中。如果不使用 -o 选项，那么将生成名为 main.i 的预处理文件。

4) 将目标文件作为输入文件，将可执行文件作为输出文件：

```
$ gcc -c func.c main.c
$ gcc func.o main.o -o app.out
```

第一条命令只编译不链接，将生成 func.o 和 main.o 两个目标文件。第二条命令将生成的两个目标文件生成最终的可执行文件 app.out。如果不使用 -o 选项，那么将生成名字为 a.out 的可执行文件。

< 上一页 下一页 >

所有教程

算法 Python爬虫 C语言入门 C语言编译器 C语言项目案例 数据结构
多线程 链接库 C++ STL C++11 socket GCC GDB Makefile
OpenCV Qt教程 Unity 3D UE4 游戏引擎 Python Python并发编程
TensorFlow Django NumPy Linux Shell Java教程 设计模式
Java Swing Servlet教程 JSP教程 JSTL Struts2 Maven Nexus
Spring Spring MVC Spring Boot Spring Cloud Hibernate Mybatis
MySQL教程 MySQL函数 NoSQL Redis常用命令手册 HBase MongoDB
Go语言 C# MATLAB JavaScript Bootstrap HTML CSS PHP
汇编语言 TCP/IP vi命令 Android教程 区块链 Docker 大数据
云计算 推荐阅读 编程笔记 资源下载 VIP视频 一对一答疑 关于我们