

## GCC生成动态链接库（.so文件）：-shared和-fPIC选项

Linux 下动态链接库（shared object file，共享对象文件）的文件后缀为 `.so`，它是一种特殊的目标文件（object file），可以在程序运行时被加载（链接）进来。使用动态链接库的优点是：程序的可执行文件更小，便于程序的模块化以及更新，同时，有效内存的使用效率更高。

### GCC 生成动态链接库

如果想创建一个动态链接库，可以使用 GCC 的 `-shared` 选项。输入文件可以是源文件、汇编文件或者目标文件。

另外还得结合 `-fPIC` 选项。`-fPIC` 选项作用于编译阶段，告诉编译器产生与位置无关代码（Position-Independent Code）；这样一来，产生的代码中就没有绝对地址了，全部使用相对地址，所以代码可以被加载器加载到内存的任意位置，都可以正确的执行。这正是共享库所要求的，共享库被加载时，在内存的位置不是固定的。

例如，从源文件生成动态链接库：

```
$ gcc -fPIC -shared func.c -o libfunc.so
```

从目标文件生成动态链接库：

```
$ gcc -fPIC -c func.c -o func.o
$ gcc -shared func.o -o libfunc.so
```

`-fPIC` 选项作用于编译阶段，在生成目标文件时就得使用该选项，以生成位置无关的代码。

### GCC 将动态链接库链接到可执行文件

如果希望将一个动态链接库链接到可执行文件，那么需要在命令行中列出动态链接库的名称，具体方式和普通的源文件、目标文件一样。请看下面的例子：

```
$ gcc main.c libfunc.so -o app.out
```

将 `main.c` 和 `libfunc.so` 一起编译成 `app.out`，当 `app.out` 运行时，会动态地加载链接库 `libfunc.so`。

当然，必须要确保程序在运行时可以找到这个动态链接库。你可以将链接库放到标准目录下，例如

/usr/lib，或者设置一个合适的环境变量，例如 LIBRARY\_PATH。不同系统，具有不同的加载链接库的方法。

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

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2020 biancheng.net, 陕ICP备15000209号

