

## GCC -l选项：手动添加链接库

[< 上一页](#)[下一页 >](#)

链接器把多个二进制的目标文件（object file）链接成一个单独的可执行文件。在链接过程中，它必须把符号（变量名、函数名等一些列标识符）用对应的数据的内存地址（变量地址、函数地址等）替代，以完成程序中多个模块的外部引用。

而且，链接器也必须将程序中所用到的所有C标准库函数加入其中。对于链接器而言，链接库不过是一个具有许多目标文件的集合，它们在一个文件中以方便处理。

当把程序链接到一个链接库时，只会链接程序所用到的函数的目标文件。在已编译的目标文件之外，如果创建自己的链接库，可以使用 `ar` 命令。

标准库的大部分函数通常放在文件 `libc.a` 中（文件名后缀 `.a` 代表“achieve”，译为“获取”），或者放在用于共享的动态链接文件 `libc.so` 中（文件名后缀 `.so` 代表“share object”，译为“共享对象”）。这些链接库一般位于 `/lib/` 或 `/usr/lib/`，或者位于 [GCC](#) 默认搜索的其他目录。

当使用 GCC 编译和链接程序时，GCC 默认会链接 `libc.a` 或者 `libc.so`，但是对于其他的库（例如非标准库、第三方库等），就需要手动添加。

令人惊讶的是，标准头文件 `<math.h>` 对应的数学库默认也不会被链接，如果没有手动将它添加进来，就会发生函数未定义错误。

GCC 的 `-l` 选项（小写的 L）可以让我们手动添加链接库。下面我们编写一个数学程序 `main.c`，并使用到了 `cos()` 函数，它位于 `<math.h>` 头文件。

```
01. //main.c
02. #include <stdio.h>      /* printf */
03. #include <math.h>       /* cos */
04. #define PI 3.14159265
05. int main ()
06. {
07.     double param, result;
08.     param = 60.0;
09.     result = cos ( param * PI / 180.0 );
10.     printf ("The cosine of %f degrees is %f.\n", param, result );
11.     return 0;
```

```
12. }
```

如果我们不使用 `-l` 选项:

```
[root@bogon demo]# gcc main.c
/tmp/ccYfkZJk.o: In function `main':
main.c:(.text+0x34): undefined reference to `cos'
collect2: ld returned 1 exit status
```

显然, GCC 编译器无法找到 `cos()` 这个函数。为了编译这个 `main.c`, 必须使用 `-l` 选项, 以链接数学库:

```
[root@bogon demo]# gcc main.c -o main.out -lm
```

数学库的文件名是 `libm.a`。前缀 `lib` 和后缀 `.a` 是标准的, `m` 是基本名称, GCC 会在 `-l` 选项后紧跟着的基本名称的基础上自动添加这些前缀、后缀, 本例中, 基本名称为 `m`。

在支持动态链接的系统上, GCC 自动使用在 Darwin 上的共享链接库 `libm.so` 或 `libm.dylib`。

## 链接其它目录中的库

通常, GCC 会自动在标准库目录中搜索文件, 例如 `/usr/lib`, 如果想链接其它目录中的库, 就得特别指明。有三种方式可以链接在 GCC 搜索路径以外的链接库, 下面我们分别讲解。

1) 把链接库作为一般的目标文件, 为 GCC 指定该链接库的完整路径与文件名。

例如, 如果链接库名为 `libm.a`, 并且位于 `/usr/lib` 目录, 那么下面的命令会让 GCC 编译 `main.c`, 然后将 `libm.a` 链接到 `main.o`:

```
[root@bogon demo]# gcc main.c -o main.out /usr/lib/libm.a
```

2) 使用 `-L` 选项, 为 GCC 增加另一个搜索链接库的目录:

```
[root@bogon demo]# gcc main.c -o main.out -L/usr/lib -lm
```

可以使用多个 `-L` 选项, 或者在一个 `-L` 选项内使用冒号分割的路径列表。

3) 把包括所需链接库的目录加到环境变量 `LIBRARYPATH` 中。

< 上一页

下一页 >

