



焯奕

[博客园](#)
[首页](#)
[新随笔](#)
[联系](#)
[订阅](#)
[管理](#)

Linux 编译运行查找头文件和库的顺序

前言

linux中在使用gcc进行编译时，可能会出现找不到相应库或头文件的情况，往往让人十分头疼。因此，此文描述了库和头文件的查找顺序和一些注意事项，希望能帮助大家在出错时能够快速定位和解决。

头文件

gcc在编译时按照如下顺序寻找所需要的头文件：

- 1.先搜索当前目录（**这里注意，只有用 `#include "headfile.h"` 时才会搜索当前目录**）
- 2.接着搜索 `-I` 指定的目录
- 3.然后找gcc的环境变量 `C_INCLUDE_PATH`, `CPLUS_INCLUDE_PATH`, `OBJC_INCLUDE_PATH`
- 4.再找内定目录: `/usr/include` , `/usr/local/include`
- 5.最后找gcc的一系列自带目录，如：

```
CPLUS_INCLUDE_PATH=/usr/lib/gcc/x86_64-redhat-linux/4.8.5/include
```

库文件

gcc在编译时按照如下顺序寻找所需要的库文件：

- 1.gcc会去找 `-L` 指定的目录
- 2.再找gcc的环境变量 `LIBRARY_PATH`
- 3.再找内定目录

- `/lib`和`/lib64`
- `/usr/lib` 和`/usr/lib64`
- `/usr/local/lib`和`/usr/local/lib64`

这是当初compile gcc时写在程序内的

这里有两个问题：

1.默认情况下，gcc编译时只会查找相应的头文件，而不会连接具体的lib。也就是说只要include设置完全，就可以编译通过。它没有进一步检查include中的类和函数有没有实现，而是在运行时才开始查找。所以就会经常发生编译可以通过，但运行时却无法运行，因为在运行时它找不到相关类或者函数的实现。

这时，使用 `-Wl,--no-undefined` 参数，如果使用了include文件，连接器却找不到相应的实现，就会产生错误提示。

2.编译时默认不查找当前目录，需要使用 `-L ./` 指定，例如

```
[lei@ ~]$ gcc -o client client.c -lssl -lcrypto
/tmp/cc1kCFRW.o: In function `main':
client.c:(.text+0x130): undefined reference to `OPENSSL_init_ssl'
client.c:(.text+0x13f): undefined reference to `OPENSSL_init_crypto'
client.c:(.text+0x14e): undefined reference to `OPENSSL_init_ssl'
client.c:(.text+0x153): undefined reference to `TLS_client_method'
collect2: error: ld returned 1 exit status
[lei@ ~]$ gcc -o client client.c -L ./ -lssl -lcrypto
[lei@ ~]$ ls
CAcert.pem  encode.c  libcrypto.a  libcrypto.so.1.1  libssl.so  pkgconfig  service
client.c    engines-1.1  libcrypto.so  libssl.a  libssl.so.1.1  privkey.pem  service.c
```

运行时动态库的搜索路径

动态库的搜索路径搜索的先后顺序是：

随笔 - 39 文章 - 1 评论 - 2 阅读 - 64084

| 2021年9月 | | | | | | |
|---------|----|----|----|----|----|----|
| < | 日 | 一 | 二 | 三 | 四 | 五 |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

搜索

最新随笔

- 1.web服务部署工具使用介绍
- 2.Clang Static Analyzer - 静...
- 3.MySQL中character set与c...
- 4.Linux 编译运行查找头文件...
- 5.git入门及使用记录
- 6.CentOS7通过yum安装gdb...
- 7.win10子系统设置开机启动...
- 8.STL中map的使用
- 9.C++逆序输出字符串
- 10.函数指针和指针函数

我的标签

C++(26)
summary(23)
algorithm(11)
数据结构(5)
算法(4)
linux(3)
数据库(3)
工具(2)
centos7(2)
剑指offer(2)
更多

积分与排名

积分 - 42390
排名 - 29834

随笔分类

algorithm(3)
C++编程(11)
C++总结(12)
CentOS(1)
Crypto(1)
git(1)
Linux(1)
Python(1)
summary(3)
工具(2)
数据结构(4)
数据库(3)

随笔档案

- 1.编译目标代码时指定的动态库搜索路径；这是通过gcc的参数 `-Wl,-rpath=` 指定。当指定多个动态库搜索路径时，路径之间用冒号 `:` 分隔
- 2.环境变量 `LD_LIBRARY_PATH` 指定的动态库搜索路径
- 3.配置文件 `/etc/ld.so.conf` 中指定的动态库搜索路径
- 4.默认的动态库搜索路径 `/lib` , `/usr/lib`

注意：

- 1.动态库搜寻路径并不包括当前目录，所以当即使可执行文件和其所需的so文件在同一文件夹，也会出现找不到问题
- 2.一般不推荐直接修改环境变量，而是修改 `/etc/ld.so.conf`，将相应的路径添加上，然后ldconfig一下就好
- 3.ldconfig做的这些东西都与运行程序时有关，跟编译时一点关系都没有，编译的时候还是该加 `-L` 就得加，不要混淆了
- 4.往 `/lib` 和 `/usr/lib` 里面lib，是不用修改 `/etc/ld.so.conf` 的，但是完了之后要调一下ldconfig(很重要)，不然这个lib会找不到。而往其他目录加lib，需要修改 `/etc/ld.so.conf`，并且要ldconfig一下。
- 5.如果确实需要改环境变量，则有以下几种情况：

- 临时修改（关闭shell后失效）：若是权限不够（无法更改 `/etc/ld.so.conf`）或只是想临时改一下环境变量用于测试，则可以使用export，就是export一个全局变量 `LD_LIBRARY_PATH`，然后运行程序的时候就会去这个目录中找lib。如：命令行执行：
`export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/user/abc/lib`

一般来讲这只是一中临时的解决方案，在没有权限或临时需要的时候使用。

- 修改作用于当前用户：在 `.bashrc` 中设置或者在 `./bash_profile` 设置，这对当前用户生效。记得 `source ~/.bashrc` 或 `source ~/.bash_profile`
- 修改作用于所有用户：在 `/etc/profile` 中设置或在 `/etc/profile.d/` 中创建一个自定义的shell(`**sh`)脚本，一般推荐使用后者。原因如下：

It's NOT a good idea to change this file unless you know what you are doing.
It's much better to create a custom.sh shell script in `/etc/profile.d/` to make custom changes to your environment, as this will prevent the need for merging in future updates.

翻译即：除非您知道自己在做什么，否则更改此文件不是一个好主意。最好在 `/etc/profile.d/` 中创建一个自定义的shell(`**sh`)脚本，以对环境进行自定义更改，因为这样可以避免在将来的更新中合并。

- 最后是关于环境变量的写法：

如：`export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/user/abc/lib`
或者这样写：`export LD_LIBRARY_PATH="{LD_LIBRARY_PATH}:/home/user/abc/lib"`

要用 `:` 隔开，注意要加上 `$LD_LIBRARY_PATH`，避免之前存在的路径失效，在前则先搜索，在后则后搜索

参考

- 1.关于开发中常见的编译器技巧
- 2.LD_LIBRARY_PATH环境变量设置及Linux动态库查找方法

感谢阅读，如有问题，请批评指正，谢谢。

分类: [Linux](#)

标签: [linux](#)

好文要顶

关注我

收藏该文



焯奕

关注 - 13

粉丝 - 3

+加关注

0

0

2020年11月(2)
2020年10月(1)
2020年3月(4)
2020年2月(1)
2019年10月(3)
2019年9月(24)
2019年1月(2)
2018年12月(2)

文章分类

centos7(1)

相册

temp(2)

最新评论

1. Re:C++中的大数乘的实现
很棒

--coolsunxu

2. Re:C++对数组进行复制
总结得非常详细

--杨杰刚

阅读排行榜

- 1. C++对数组进行复制(1881...
- 2. C++输入带空格的字符串(...
- 3. CentOS7通过yum安装gd...
- 4. C++逆序输出字符串(3600)
- 5. 求一个整数的所有因子(32...

评论排行榜

- 1. C++对数组进行复制(1)
- 2. C++中的大数乘的实现(1)

推荐排行榜

- 1. C++ 把引用作为返回值(2)
- 2. C++输入带空格的字符串(2)
- 3. Clang Static Analyzer - ...
- 4. CentOS7通过yum安装gd...
- 5. 函数指针和指针函数(1)