

patchelf 的功能以及使用 patchelf 修改 rpath 以解决动态库问题

原创 longyu_wlz 于 2020-09-12 16:25:30 发布 6882 收藏 16

版权

分类专栏: Linux 文章标签: patchelf rpath httpd 程序rpath修改 set-rpath



Linux 专栏收录该内容

2 订阅 135 篇文章

订阅专栏

在 低版本 libc 库运行高版本 libc 库编译的程序 这篇博客中我描述了使用 patchelf 来修改动态库链接器的方法，在本篇文章中，我完整的列举下 patchelf 的功能，并介绍另外一个实际的应用。

patchelf 具有的功能

运行 patchelf -h 能够得到如下信息：

```
1 syntax: patchelf
2   [--set-interpretor FILENAME]
3   [--page-size SIZE]
4   [--print-interpretor]
5   [--print-soname]          Prints 'DT_SONAME' entry of .dynamic section. Raises an error if DT_SONAME doesn't exist
6   [--set-soname SONAME]     Sets 'DT_SONAME' entry to SONAME.
7   [--set-rpath RPATH]
8   [--remove-rpath]
9   [--shrink-rpath]
10  [--allowed-rpath-prefixes PREFIXES]      With '--shrink-rpath', reject rpath entries not starting with the allowed pref:
11  [--print-rpath]
12  [--force-rpath]
13  [--add-needed LIBRARY]
14  [--remove-needed LIBRARY]
15  [--replace-needed LIBRARY NEW_LIBRARY]
16  [--print-needed]
17  [--no-default-lib]
18  [--debug]
```

```
19 | [--version]  
20 | FILENAME
```

中文翻译如下：

- 设置动态库解析器
- 设置页大小
- 设置 DT_SONAME
- 设置 rpath
- 删除 rpath
- 添加允许的 rpath 前缀
- 打印 rpath
- 强制使用 rpath
- 添加需要的动态库
- 删除需要的动态库
- 替换旧的动态库为新的
- 打印帮助信息
- 不链接默认的动态库
- 输出调试信息
- 打印版本号

从上面的功能描述中可以看到，patchelf 的主要功能与动态库解析器、rpath、动态库本身相关，可能在解决一些动态库链接程序执行的问题时能够用到。

下面是一个具体的实例。

patchelf 修改 rpath 以使用自动以目录中的动态库

最近有同事找我们帮忙解决一个动态库的问题，问题的具体情况是他编译出来的 httpd 程序一直使用的是系统默认路径中的动态库，而他的需求是要使用自定义目录中的动态库。

他尝试过设定 LD_LIBRARY_PATH 结果没有生效，就来找我们帮忙看看。

我在 [man ld.so 的翻译](#) 这篇文章中翻译了 ld.so 动态库链接器执行的过程，其中查找动态库的步骤如下：

1. 针对 ELF 格式文件，当 DT_RUNPATH 属性不存在的情况下，使用二进制程序 dynamic section 中存在的 DT_RPATH 属性指定的路径来搜索。DT_RPATH 已经被弃用。
2. 使用环境变量 LD_LIBRARY_PATH 中指定的路径来搜索。如果可执行程序设定了 setuid/setgid，这一步将被跳过。
3. 从缓存文件 /etc/ld.so.cache 中查找。如果程序在链接时使用了 -z nodeflib 选项，默认库路径中的库及那个会被跳过。安装到硬件兼容目录中的库将会比其它库优先查找。
4. 在默认的 /lib 然后时 /usr/lib 中寻找，如果程序在链接时使用了 -z nodeflib 选项，这一步将被跳过

可以看到在搜索 LD_LIBRARY_PATH 之前会先以 ELF 文件中存在的 DT_RPATH 属性中指定的路径来搜索动态库，看上去这个问题就出在这里。

确定问题

运行 readelf -a httpd 搜索与 rpath 相关的内容，果然搜索到了，发现确实**设定了这个变量的值**，并且指向**默认路径**，这就是导致 LD_LIBRARY_PATH 不能生效的原因。

确定了问题后，搜索 httpd 编译目录中的 Makefile 文件，发现 rpath 的设定是通过向编译器传参设置的，**确定问题应该是 configure 的时候没有进行某种配置**。

临时让 httpd 程序先跑起来的方法

httpd 的配置与编译过程相对复杂，要解决上面的问题可能要搞一会，这时我们想用一些更简单的方法先让 httpd 程序跑起来，这其实可以通过 patchelf 来实现。

运行如下命令，将 rpath 的只修改为自定义的动态库目录就解决了这个问题。

```
1 | patchelf --set-rpath '/home/xx/local/apr/apr/lib:/home/xx/local/apr/util/lib/' httpd
```

有没有其它的方法？

其实这个问题也可以直接删除 rpath 的设定，然后设定 LD_LIBRARY_PATH 来解决，这其实与修改 ELF 文件中的 rpath 属性的内容大同小异。

总结

我们解决问题依赖我们**掌握的知识、阅读过的书、写过的代码、运行过的 demo、做过的解决相同问题的记录**。在真正解决问题的时候，可能我们还是存在一定的欠缺，这些欠缺在我看来很多是我们对**现有的工具的大致工作原理与其提供的功能存在盲点**，其实我们不必要知道所有的细节，但是对于我们的业务范围内使用到的工具提供的功能需要有**全面的了解**，这样我们才可能能够轻松的解决一些因欠缺知识而看似困难的问题。