低版本 libc 中运行高本版 libc 库链接的程序

原创 longyu_wlz 于 2020-08-22 11:28:55 发布 1063 收藏 6 版权

文章标签： 动态库解析器 ld-x86-64 运行高版本libc库程序

问题描述

需要在低版本 libc 中运行高本版 libc 库编译的 dpdk-19.11 的 l2fwd 程序进行测试，直接运行是会失败的，需要进行一些额外的处理，主要有三种方法。

三种不同的方法

第一种方法：复制新的库与动态库解析器，使用高版本的动态库解析器执行程序

最开始，我直接复制 l2fwd 程序依赖的高版本动态库与动态库解析器，在目标机器上高版本动态库的同级目录中运行 l2fwd 程序，结果报了如下错误：

```
relocation error: /usr/lib64/libc.so.6: symbol _dl_starting_up, version GLIBC_PRIVATE not define
```

这个报错表明 l2fwd 程序仍旧使用的是目标机器上的低版本的动态库加载器来加载动态库。

可以直接使用高版本的动态库解析器运行 l2fwd 来解决这个问题，执行命令如下所示：

```
./ld-linux-x86-64.so.2 ./l2fwd -- -p0x1e
```

第二种方法：复制新的库然后模拟容器的方式，执行 chroot

这种方法相对复杂，不只要拷贝 l2fwd 程序依赖的动态库还要拷贝 /bin 目录中的一些命令，**同时注意 l2fwd 还要访问 /dev /sys /mnt/huge /proc 这些目录**，需要执行 **mount --bind** 来将这些目录挂载到新的目录中。

在宿主机上运行如下命令：

```
mount --bind /dev ./dev/
mount --bind /usr ./usr/
mount --bind /proc ./proc/
mount --bind /sys ./sys
mount --bind /mnt/huge/ ./mnt/huge/
mount --bind /mnt/ ./mnt/
mount --bind /opt ./opt
mount --bind /sys/fs/cgroup/ ./sys/fs/cgroup/
```

挂载后执行 chroot 切换根目录，然后运行 l2fwd 命令就可以了。

第三种方式：修改可执行目标文件中的动态库解析器指向的位置

动态库解析器的位置在动态链接程序中的 .interp section 中存储，可以使用 readelf 工具查看。

一个示例如下：

```
[longyu@debian-10:11:15:16] build $ readelf -a l2fwd | grep interp
  [ 1] .interp           PROGBITS         00000000000002e0  000002e0
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
   01     .interp
```

可以看到这个 l2fwd 使用的动态库解析器为 **/lib64/ld-linux-x86-64.so.2** 文件。我们可以修改这个 setction 的位置，让它指向新的位置来使用新的动态库链接器。

注意不要直接编辑目标文件，当你修改了目标文件中 interp section 的内容后，**其它 section 的地址偏移也需要改变**，而编辑器并不会自动转换，这样你将会得到一个**异常的可执行文件。**

patchelf

网上搜索了下发现可以用 patchelf 命令来完成，首先执行如下命令安装 patchelf。

```
sudo apt-get install patchelf
```

查看 manual 找到了如下与 interp 相关的子命令：

```
  [--set-interpreter FILENAME]
```

执行这个命令修改动态库解析器的位置，修改后可以正常执行。