

练习 11

动态链接

在练习 09 中,您使用了在程序启动时自动加载的共享动态链接库。在

在本练习中,您将在程序控制下直接使用 Linux 动态链接 API。这将允许您在程序运行时加载并执行代码,甚至是您的程序在编写时不知道的代码。技巧

像这样用于为应用程序提供插件或其他一些扩展接口。可扩展的应用程序可以编写为搜索、加载和执行扩展其功能的新代码模块,所有这些都不需要重新编译应用程序。

您将使用我已经为您创建的一些碎片对象,因此您需要在 EOS 上完成此练习 Linux 机器确保这些对象是兼容的。登录 EOS Linux 机器,创建一个目录,其中如果您想进行此练习,请复制我准备的入门文件。您应该能够使用以下内容卷曲命令:

```
cp /mnt/coe/workspace/csc/CSC246/001/ex11/*。
```

不要忘记最后的这个点。

您刚刚复制了一个名为 polyglot.c 的 (大部分是空的)源文件和三个名为 hello-0.so、hello-1.so 的共享对象和 hello-2.so。这些对象中的每一个都包含一个函数的机器代码,如下所示。像练习 09,此函数动态分配一个字符数组,用一个字符串填充它 (包含一条消息,如 “Hello world.”) 并返回指向该字符串的指针。

```
字符 *getMessage() {
    ...
}
```

您将完成 polyglot.c 程序,以便它能够任意数量的共享对象加载代码,例如这。共享对象列表将在命令行中给出。它会加载每个共享对象,找到一个指向它的指针 getMessage() 函数,调用 getMessage() 函数获取一个字符串,打印出字符串,然后通过释放动态分配字符串并关闭共享对象。

通过为程序提供不同的命令行参数,我们可以让它加载任何共享对象。例如,按如下方式运行程序将告诉它只使用 hello-0.so。此对象中的 getMessage() 函数返回通常,“你好世界。”信息。

```
./polyglot hello-0.so
你好世界。
```

通过在命令行列出多个共享对象,我们可以让程序使用getMessage()函数来自多个对象。hello-1.so 和 hello-2.so 共享对象返回不同语言的 hello world 消息。

```
./polyglot hello-1.so hello-2.so
```

你好世界。

你好世界。

我们甚至可以使用shell的模式匹配语法让程序使用当前的所有共享对象目录。

```
./多语言 *.so
```

你好世界。

你好世界。

你好世界。

加载共享对象

您应该已经知道如何迭代命令行参数。我们只需要添加代码即可加载使用

每个参数命名的共享对象。您需要的三个函数在 Linux 中的 libdl 库中定义。

您可以使用“man”命令（例如，“man dlopen”）阅读这些功能的联机文档。

- dlopen()

dlopen() 函数打开一个共享对象并返回一个空指针作为后续的句柄

使用对象。它需要两个参数。第一个是你想要的共享对象的路径名

打开,第二个给出额外的标志。对于第二个参数,我成功地使用了 RTLD_LAZY 或 RTLD_NOW。对于第一个参数,我必须在名称的开头加上一个“./”,所以

函数会将其解释为相对路径（文档对此有一些说明）。例如,如果“hello-0.so”是命令行参数之一,我将字符串“./hello-0.so”作为

要打开的对象的名称。

- dlsym()

成功打开其中一个共享对象后,使用dlsym()查找getMessage()的地址

象征。您需要传入从 dlopen() 返回的句柄和所需符号的名称

来自对象“getMessage”。这将返回一个空指针,但您想将其分配给更具体的指针

类型。getMessage 符号应该是指向不带参数并返回 char 指针的函数的指针。将 dlsym() 的返回值赋值给这种指针类型,就可以使用函数了

调用 getMessage() 函数的指针,获取它返回的字符串,然后打印（并释放）该字符串。

- 关闭()

完成库后,使用 `dlclose()` 释放与共享对象关联的资源并告诉操作系统,它可以从您的地址空间中取消映射。

要使用这些库函数,您需要包含 `dlfcn.h` 标头 (已包含在启动程序中)。

您还需要链接到 `dl` 库。以下内容应该可以编译您的程序:

```
gcc -Wall -std=c99 -g polyglot.c -o polyglot -ldl
```

一旦你的多语言程序开始工作,它应该能够加载和使用任何提供

`getMessage()` 函数与上述函数类似。您应该能够如上例所示运行它,使用练习中提供的三个共享对象。它甚至应该与其他共享对象一起工作,那些你没有了解您编写程序的时间 (当我们测试您的程序时,我们会尝试看看它是否有效)。

完成后,将已完成的 `polyglot.c` 程序的源代码提交到 Gradescope 作业命名为 EX11。