

Shell脚本中调用另一个Shell脚本的三种方式

主要以下几种方式:

Command	Explanation
fork	新开一个子 Shell 执行, 子 Shell 可以从父 Shell 继承环境变量, 但是子 Shell 中的环境变量不会带回给父 Shell。
exec	在同一个 Shell 内执行, 但是父脚本中 exec 行之后的内容就不会再执行了
source	在同一个 Shell 中执行, 在被调用的脚本中声明的变量和环境变量, 都可以在主脚本中进行获取和使用, 相当于合并两个脚本在执行。

第一种: fork 特点: 会生成子PID而且可重复被调用。

◆ fork 是最普通的, 就是直接在脚本里面用 `path/to/foo.sh` 来调用

◆ `foo.sh` 这个脚本, 比如如果是 `foo.sh` 在当前目录下, 就是 `./foo.sh`。运行的时候 terminal 会新开一个子 Shell 执行脚本 `foo.sh`, 子 Shell 执行的时候, 父 Shell 还在。子 Shell 执行完毕后返回父 Shell。子 Shell 从父 Shell 继承环境变量, 但是子 Shell 中的环境变量不会带回父 Shell。

1.进入 编辑:

```
[root@localhost ~]# vim liqiang-2.sh
```

2.编辑内容调用写法:

echo 在这里调用

```
./liqiang.sh
```

3.输出结果

```
[root@localhost ~]# ./liqiang-2.sh
```

在这里调用

hollo

hollo

第二种: exec 特点: exec调用一次之后的所有代码都不执行

◆ exec 与 fork 不同, 不需要新开一个子 Shell 来执行被调用的脚本. 被调用的脚本与父脚本在同一个 Shell 内执行。但是使用 exec 调用一个新脚本以后, 父脚本中 exec 行之后的内容就不会再执行了。这是 exec 和 source 的区别。

1.编辑内容调用写法:

echo 在这里调用

```
./liqiang.sh
```

echo 这里是exec函数调用

```
exec ./liqiang-3.sh
```

echo 这里是source

```
source liqiang-4.sh
```

echo 在这里调用

```
./liqiang.sh
```

echo 这里是exec函数调用

```
exce liqiang-3.sh
```

echo 这里是source

```
source liqiang-4.sh
```

echo 在这里调用

```
./liqiang.sh
```

echo 这里是exec函数调用

```
exce liqiang-3.sh
```

echo 这里是source

```
source liqiang-4.sh
```

2.输出结果: 特点在于执行了第一个exec之后下面的就再也不执行了。

```
[root@localhost ~]# ./liqiang-2.sh
```

在这里调用

hollo

hollo

这里是exec函数调用

看看这里调用几次
调用一次就对了

第三种：source特点不会生成子PID也就是子进程，他就行把被用的脚本拷贝到当前shell脚本中执行，可以重复被调用。

◆与 fork 的区别是不新开一个子 Shell 来执行被调用的脚本，而是在同一个 Shell 中执行. 所以被调用的脚本中声明的变量和环境变量，都可以在主脚本中进行获取和使用。

1.编辑内容调用写法：

```
echo 在这里调用
./liqiang.sh
echo 这里是exec函数调用这次被注释掉了
#exec ./liqiang-3.sh
echo 这里是source
source liqiang-4.sh
echo 在这里调用
./liqiang.sh
echo 这里是exec函数调用被注释掉了
#exce liqiang-3.sh
echo 这里是source
source liqiang-4.sh
echo 在这里调用
./liqiang.sh
echo 这里是exec函数调用
exec liqiang-3.sh
echo 这里是source
source liqiang-4.sh
```

◆2.执行结果：

```
[root@localhost ~]# ./liqiang-2.sh
在这里调用
hollo
hollo
这里是exec函数调用这次被注释掉了
这里是source
看看这里调用几次
没有限制就对了
在这里调用
hollo
hollo
这里是exec函数调用被注释掉了
这里是source
看看这里调用几次
没有限制就对了
在这里调用
hollo
hollo
这里是exec函数调用#之后的没被调用了
```

注意：

◆source方式的结果是两者在同一进程里运行。该方式相当于把两个脚本先合并再运行。

◆给多个脚本赋权限

chmod a+x liqiang-2.sh liqiang-3.sh liqiang-4.sh