**实验一：**

**Environment Variable and Set-UID Program Lab**

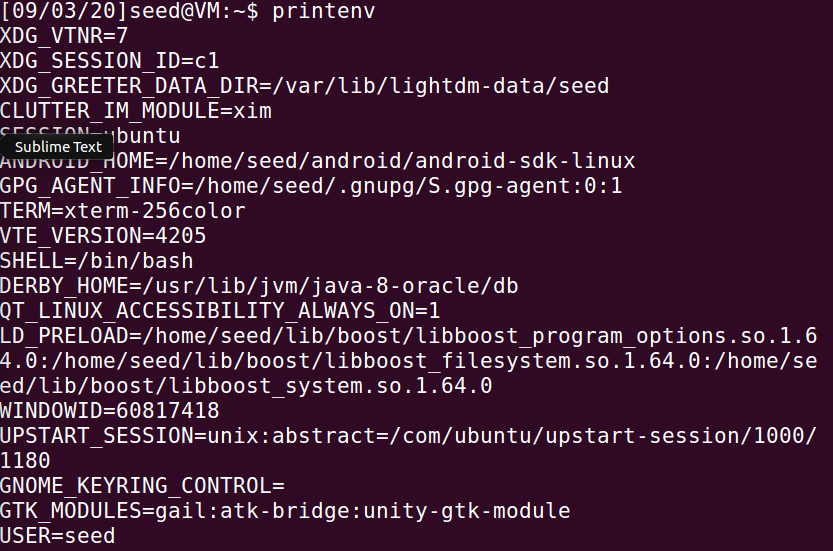
**57117101 焦典**

**Task1：Manipulating Environment Variables**

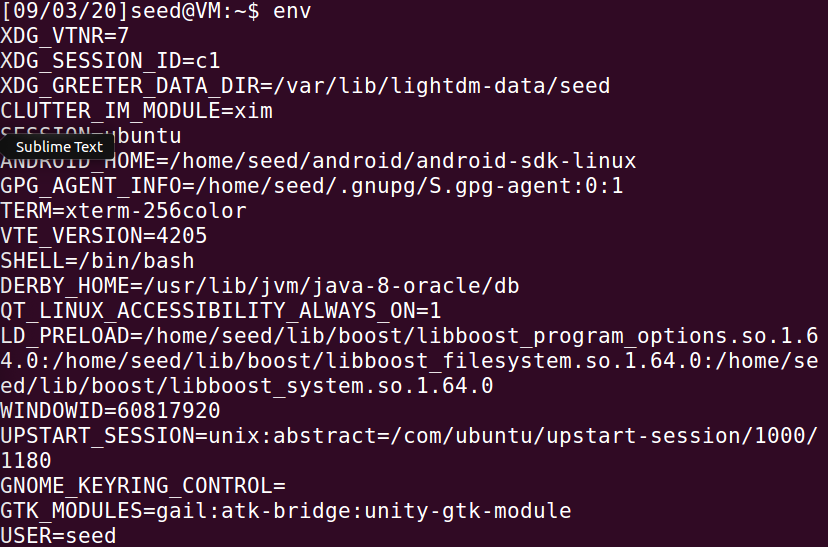
1.实验内容：输入printenv或env命令输出环境变量；输入export或unset设置或取消环境变量。

2.实验步骤及结果

（1）输入printenv命令：



输入env命令：

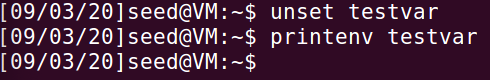


（2）输入export设置环境变量：先设置一个测试的环境变量testvar，然后输入printenv查看，发现设置成功。

4B27B8015E18E03261F9366755D6B34F

3FE8FD29EEBAA8574184642F15689C91

输入unset取消环境变量：输入unset取消环境变量，通过printenv发现取消成功。



**Task2：Passing Environment Variables from Parent Process to Child Process**

1.实验内容：观察父进程和子进程的环境变量

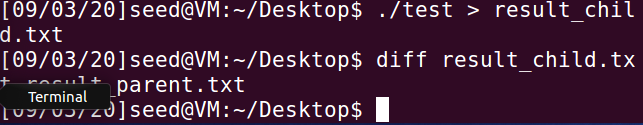
2.实验步骤及结果

（1）编译代码，并将结果输出到父进程的txt文件中：

D3AD3AAA9BC336B60C9CA34FD9F59C3E

99DEA3308A4FA5A3DB2E1A02EFAAA1EF

（2）修改代码后，重新编译后，将结果输出到子进程的txt文件中，最后用diff命令将两个文件进行比较，发现无输出，证明父子进程具有相同的环境变量。



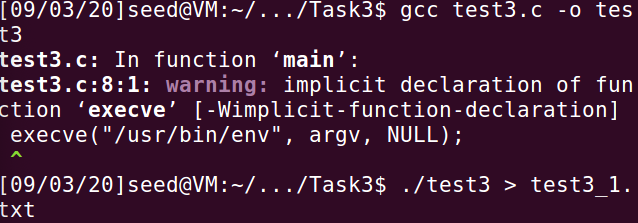
3.实验结论：利用fork系统调用的子进程和父进程的环境变量相同。

**Task3：Environment Variables and execve()**

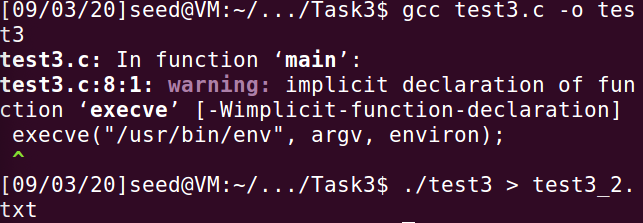
1.实验内容：观察execve是如何影响被执行程序的环境变量的，并改变第三个参数，比较结果。

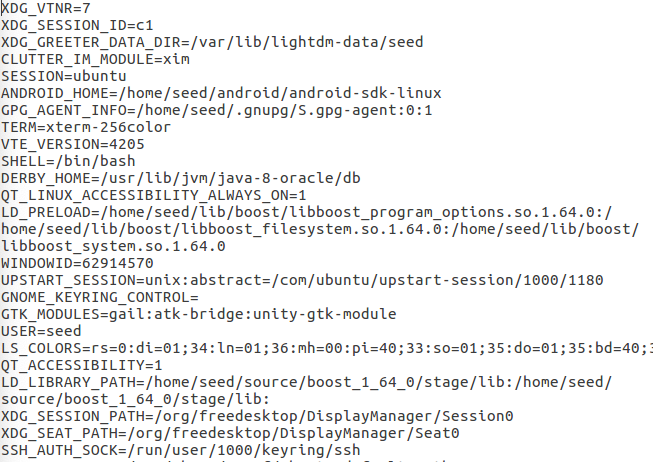
2.实验步骤及结果

（1）编译程序，将结果输出到txt文件中：



修改execve("/usr/bin/env", argv, environ); 重新编译，发现，当参数为null时，输出结果为空，当参数为environ时，输出结果如下图：





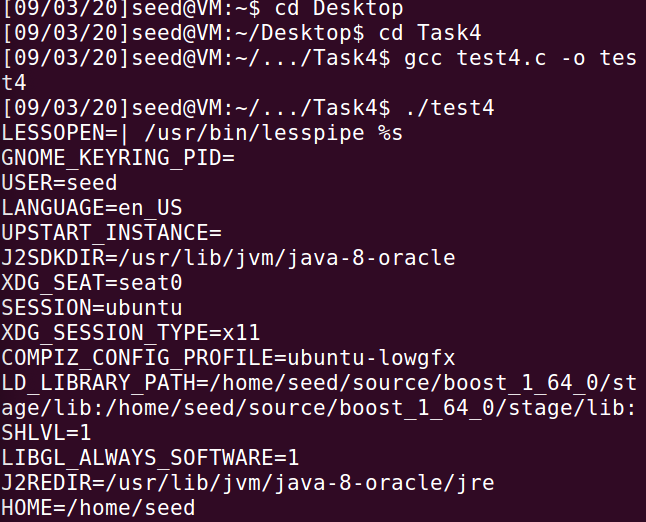
3.实验结论：新进程会默认继承父进程的环境变量，但是涉及到execve时，会根据参数的不同而改变自己的环境变量。

**Task4：Environment Variables and system()**

1.实验内容：观察system是如何影响被执行程序的环境变量的。

2.实验步骤及结果

（1）编译代码，输出结果：



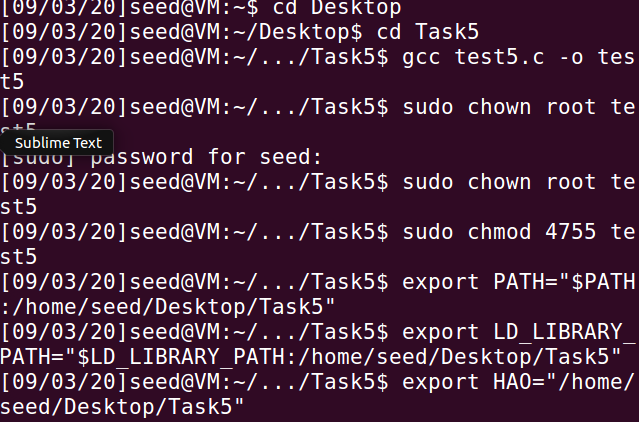
3.实验结论：通过system执行的新程序与调用程序具有相同的环境变量。

**Task5：Environment Variable and Set-UID Programs**

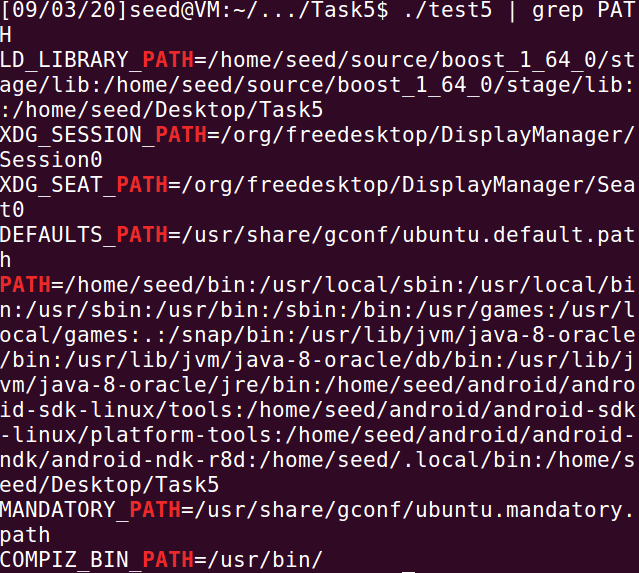
1.实验内容：将程序设置为**Set-UID程序，观察其环境变量。**

2.实验步骤及结果

（1）编译程序，将所有者改变成root，变成**Set-UID程序，再添加环境变量path并赋值。**



（2）运行修改后的程序，得到结果如下。



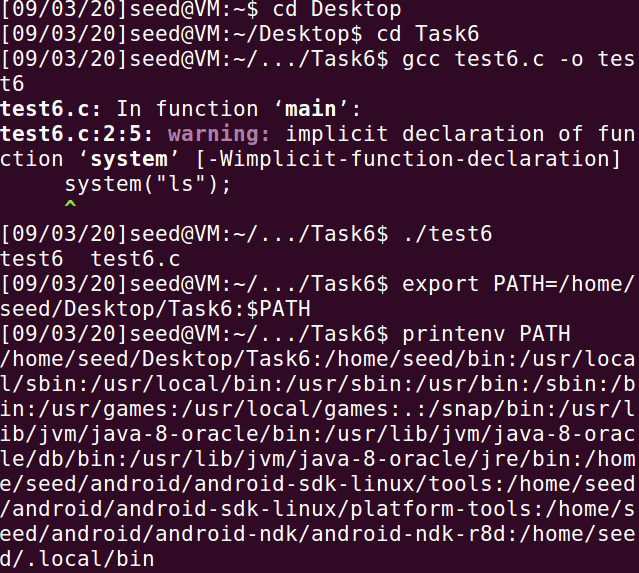
3.实验结论：**Set-UID程序中的环境变量不完全是用户所定义的环境变量。**

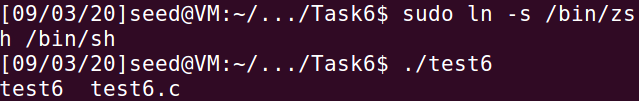
**Task6：The PATH Environment Variable and Set-UID Programs**

1.实验内容：观察path环境变量改变后的结果。

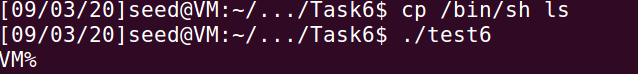
2.实验步骤及结果

（1）编译代码，修改环境变量path，再输出其结果：





（2）对系统进行处理后，发现不在需要sudo命令也可以拥有root权限：



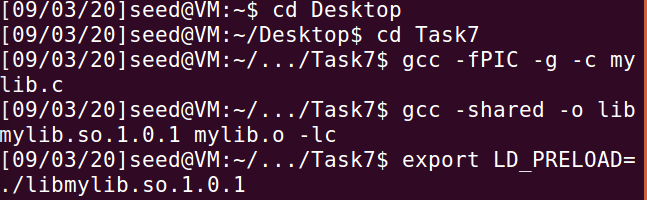
3.实验结论：Set-UID 程序的子程序是具有 root 权限的。

**Task7：The LD\_PRELOAD Environment Variable and Set-UID Programs**

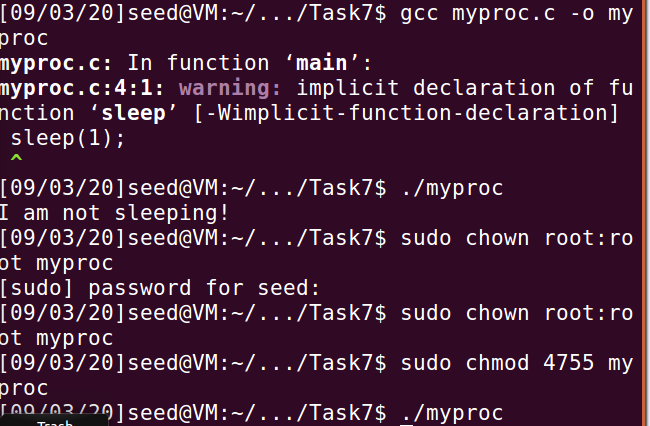
1.实验内容

2.实验步骤及结果

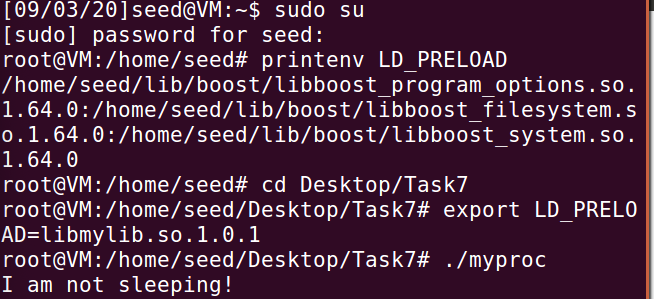
（1）：编译代码，建立动态链接库，设置环境变量LD-PRELOAD



（2）将myproc变为Set-UID 程序，再次运行，发现无显示，说明此时myproc程序对应的 LD\_PRELOAD环境变量已经不是之前seed中的值了。



（3）重新进入root权限后，发现运行结果回到了原来的值。



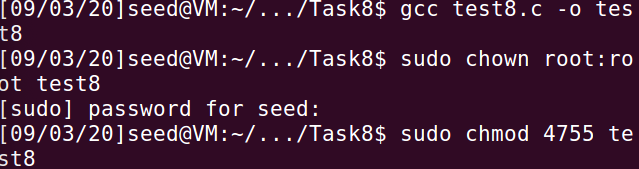
3.实验结论：只有当ruid和euid相同时，环境变量才会有效，否则不产生效果。

**Task8：Invoking External Programs Using system() versus execve()**

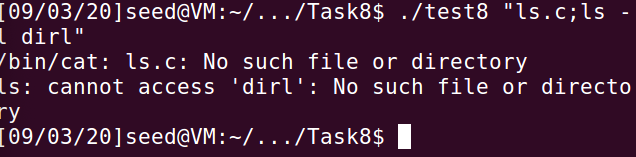
1.实验内容：观察system和execve的不同之处。

2.实验步骤及结果

（1）编译程序，并将其设置为Set-UID 程序



（2）发现无法越权操作



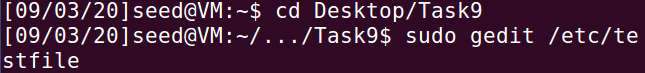
3.实验结论：execve比system的方法更加安全。

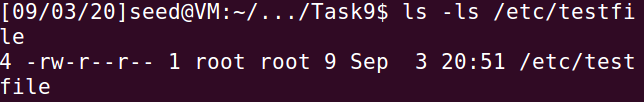
**Task9：Capability Leaking**

1.实验内容：当攻击者输入恶意语句，观察Capability Leaking的情况。

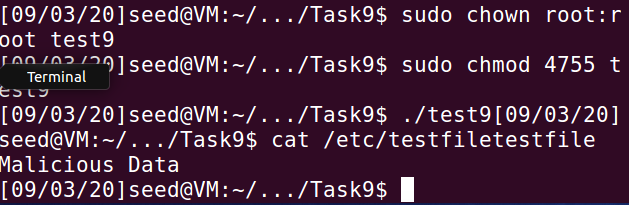
2.实验步骤及结果

（1）编译程序，在/etc下建立新的文件testfile，并写入“testfile”的内容，可见testfile文件的权限为root：





（2）编译程序，并将其设置为Set-UID 程序后，再次运行，可见此时打印的内容为修改过后的内容。

0D6633ABA0180FDA76AA1D90E8A25EA8

3.实验结论：执行程序后没有撤销特权就还可以具有root权限，造成对testfile文件的修改。