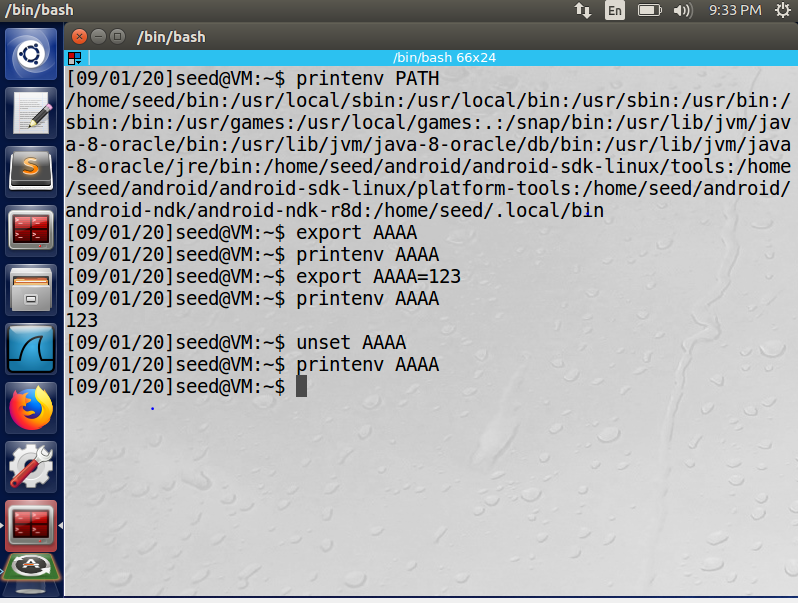
Lab 1

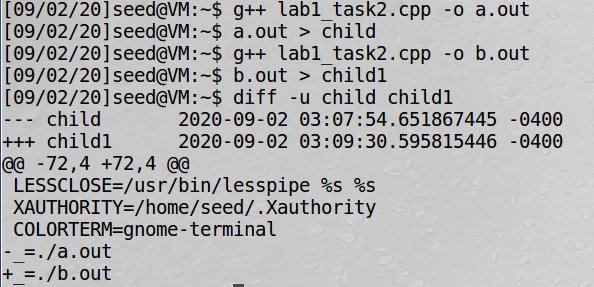
57118111 王雯

实验目的：实验为了了解环境变量是如何影响程序的和系统的行为。环境变量是一组动态命名值，可以影响这种方式正在运行的进程将在计算机上运行。大多数操作系统都使用它们。在这个实验中,我们将了解环境变量是如何工作的，以及它们如何从父进程传播到父进程儿童，以及他们如何影响系统/程序行为。我们特别感兴趣的是环境如何变量影响Set-UID程序的行为，而Set-UID程序通常是特权程序。

Task 1: Manipulating Environment Variables

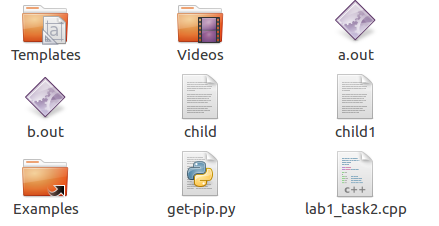


Task 2: Passing Environment Variables from Parent Process to Child Process



我们首先根据示例的代码第一步执行后，将可执行文件a.out中的字符放入child 文件中；第二步将子进程的printenv()注释掉后，将可执行文件b.out中的字符放入child1文件中；

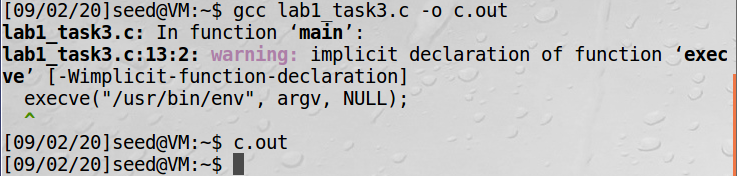
用diff 命令找出两个文件的不同之处。



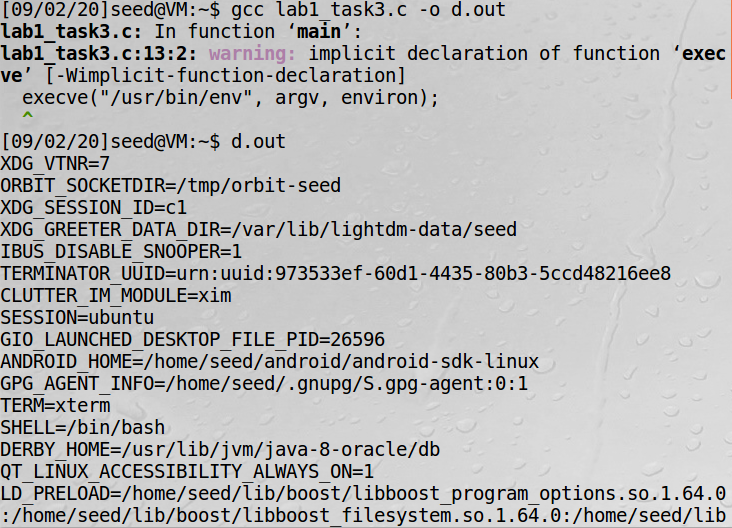
结论：我们可以从该任务得出结论，子进程使用fork()函数，会完全继承父进程的环境变量。

Task 3: Environment Variables and execve()

第一步：源码是execve()的函数中传递了NULL，子进程传递什么都没有，所以在输出c.out是无任何输出

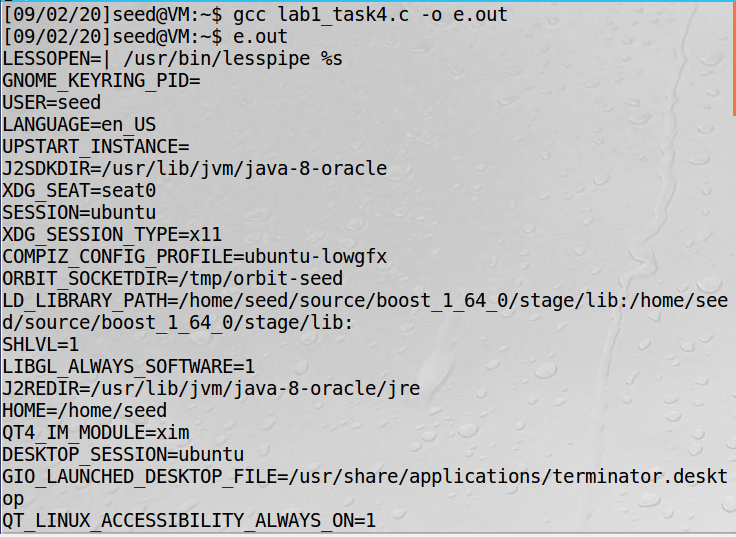


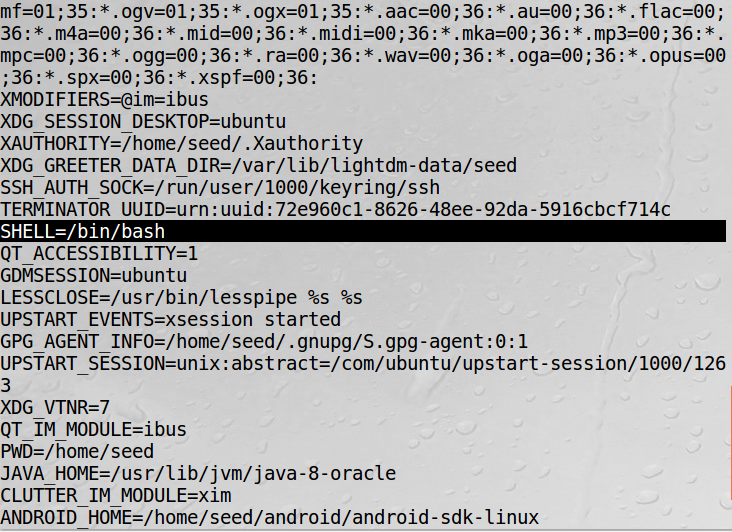
第二步：源码中execve()的函数中传递了自己的environ，所以在d.out的输出中，如下面截图所示。



结论：execve()函数在对传递不同的环境变量时的情况不同。

Task 4: Environment Variables and system()





System通过execl调用shell，会自动把环境变量传递过去。

Task 5: Environment Variable and Set-UID Programs

实验源码：

#include <stdio.h>

#include <stdlib.h>

extern char \*\*environ;

int main()

{

int i = 0;

while (environ[i] != NULL)

{

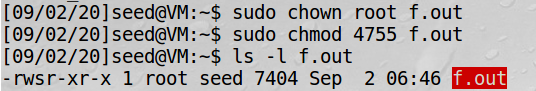
printf("%s\n", environ[i]);

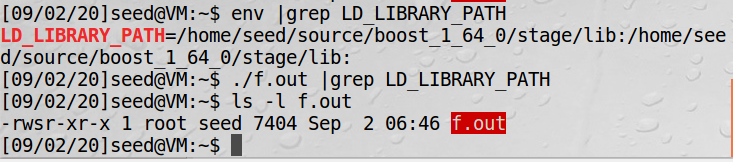
i++;

}

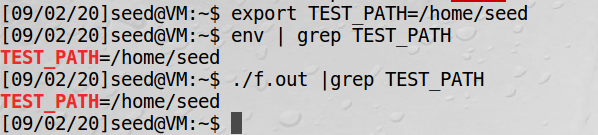
return 0;

}

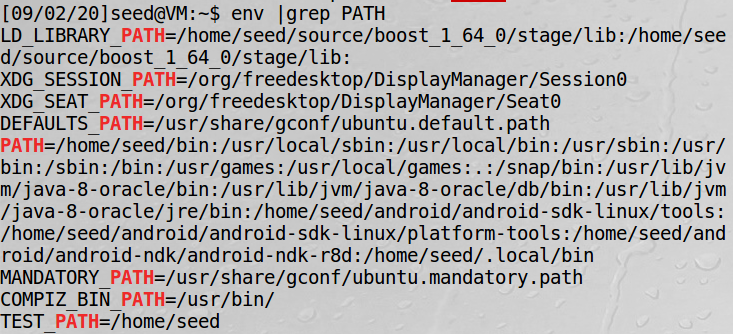




通过上图的实验结果，可得并没有继承父进程的LD\_LIBRARY\_PATH环境变量



通过上图的实验结果，可得会继承自定义的TEST\_PATH环境变量

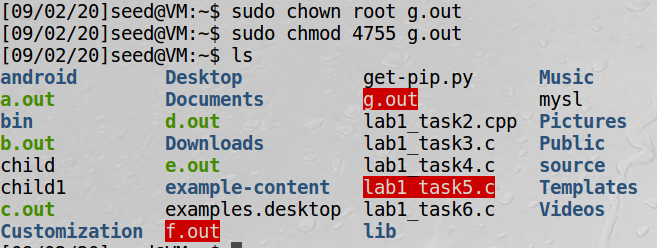




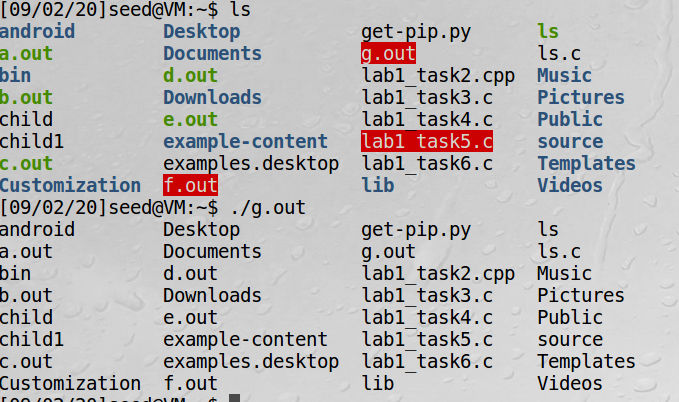
通过上图的实验结果,可得会继承PATH环境变量

Task 6: The PATH Environment Variable and Set-UID Programs

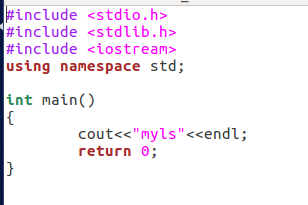
编译程序后将，g.out设置为set-uid程序



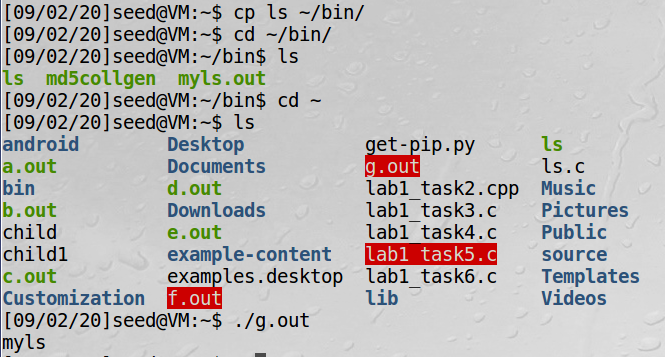
第一次运行



下图为自己编译的myls程序：

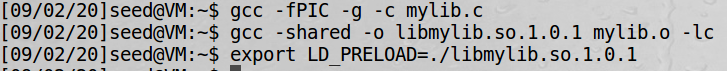


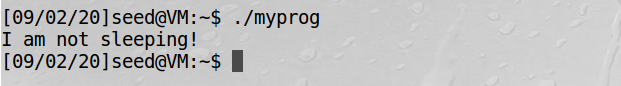
运行，发现运行的是自己的myls的程序



Task 7: The LD PRELOAD Environment Variable and Set-UID Programs

根据题意：先编译动态链接库，之后运行，发现链接到自己写的sleep函数中



1、Make myproga regular program, and run it as a normal user



2、Make myproga Set-UID root program, and run it as a normal user.

将myprog的设置为set-uid程序

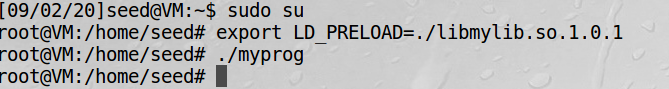




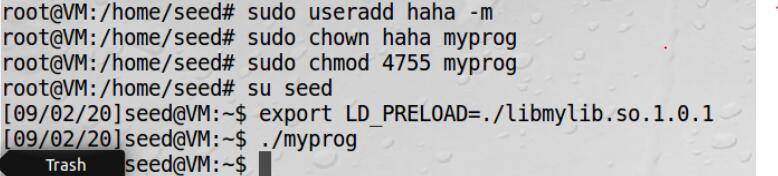
发现执行的是系统自带的sleep。

原因是，动态链接器的防御措施，特权程序类型的子进程不继承父进程的动态链接库LD\_\*环境变量。

3、Make myproga Set-UID root program, export the LD PRELOAD environment variable again inthe root account and run it.

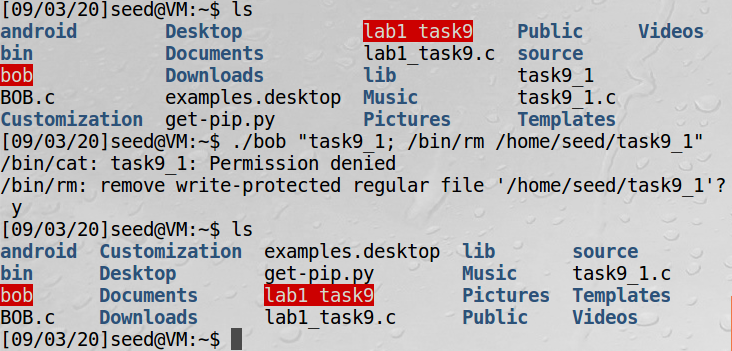


4、Make myproga Set-UID user1 program (i.e., the owner is user1, which is another user account),export the LD PRELOAD environment variable again in a different user’s account (not-root user) andrun it.

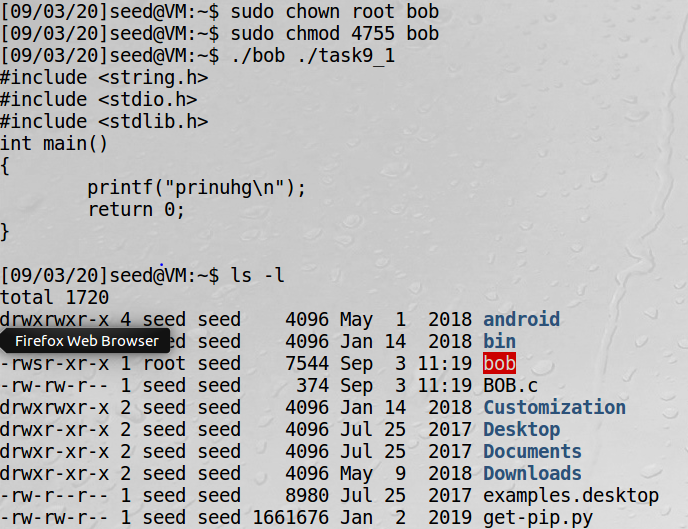


2.8 Task 8: Invoking External Programs Using system() versus execve()

1、正常情况下普通用户不可以删除root目录下的文件。但可以强制删除seed用户下，root创建的文件。



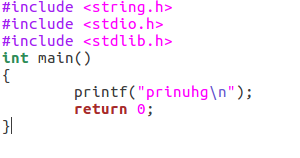
2、如果更改为execve没有被删除

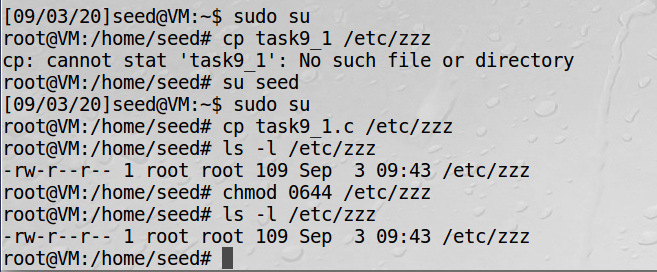


2.9 Task 9: Capability Leaking

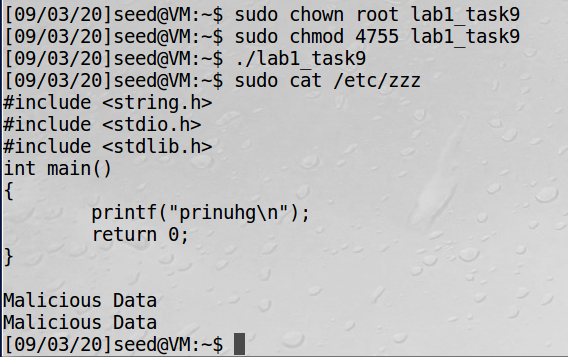
创建/etc/zzz文件

将自己的写的task9\_1文件复制到/etc/zzz文件中

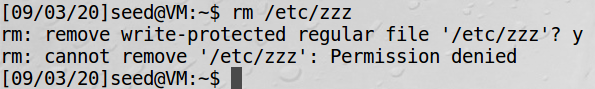




设置特权程序，并运行



由上图可知，文件已被修改



但在正常情况下用户没有权限删除文件。

实验总结：通过这次实验，我们对环境变量与set-uid程序的了解更深一步，通过实验，更为形象化的对环境变量的作用，功能，以及一些潜在传递规则都有了更深一步的了解