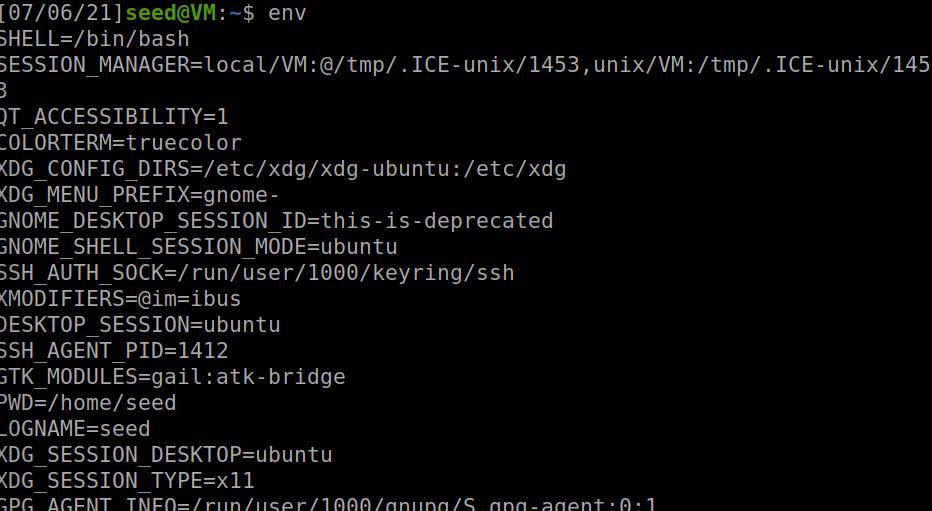
操作系统实验报告

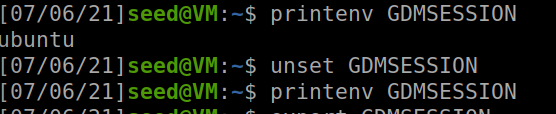
高昂57119123

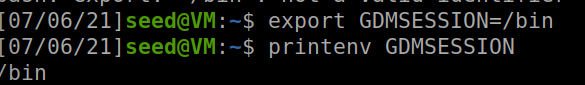
Task1

使用env打印环境变量



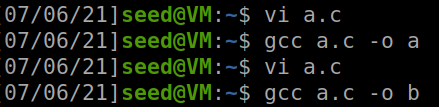
对环境变量进行修改，删除以及打印

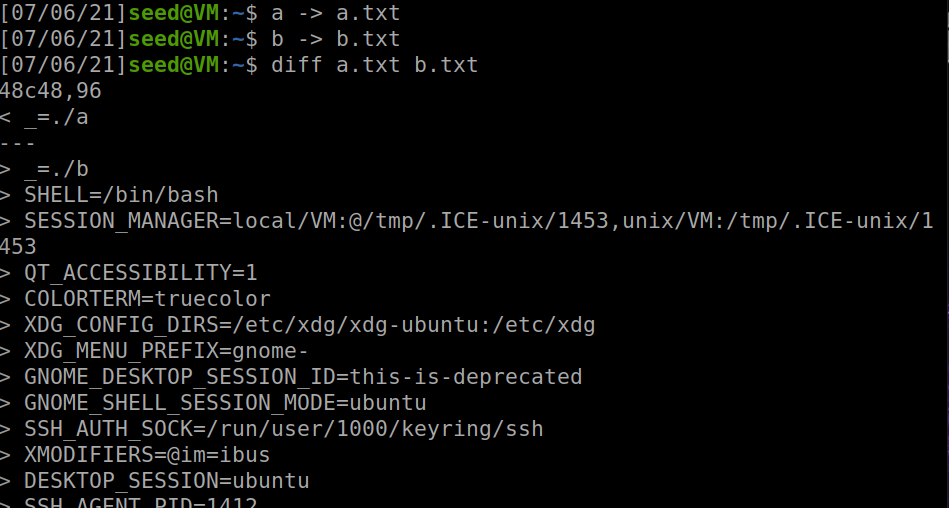




Task2

将代码原输出以及修改后输出，写入a.txt以及b.txt将其用diff比较发现，两者仅运行程序不同，即子进程会继承父进程的环境变量



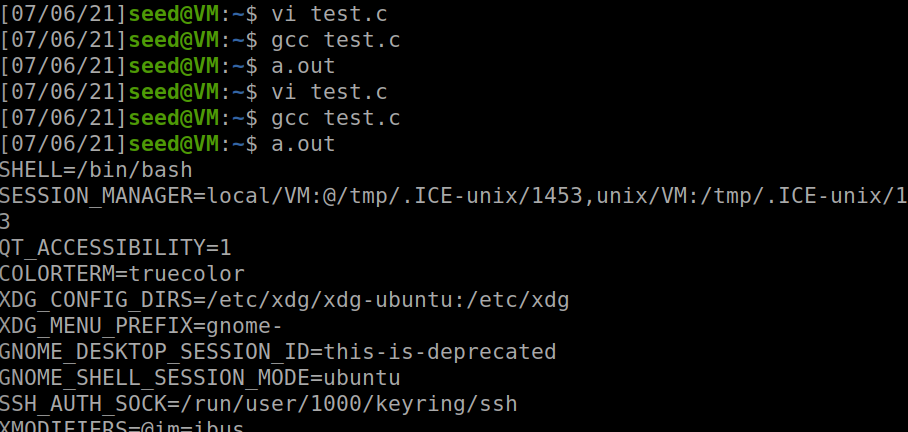


Task3

按步骤进行操作后，发现，第一次调用execve不会有输出，但是第二次调用execve函数有输出。

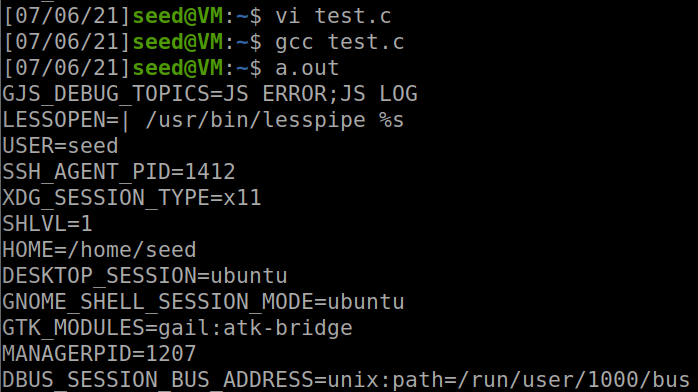
execve()用来执行参数filename字符串所代表的文件路径，第二个参数是利用数组指针来传递给执行文件，并且需要以空指针(NULL)结束，最后一个参数则为传递给执行文件的新环境变量数组。

execve函数产生的子程序不能继承环境变量等参数。因此第一次没有输出是因为没有传入环境变量，第二次有输出是因为传入了environ环境变量因此有输出。



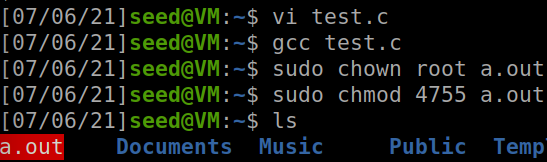
Task4

按照步骤运行后，发现使用system调用命令时，会继承当前程序的环境变量

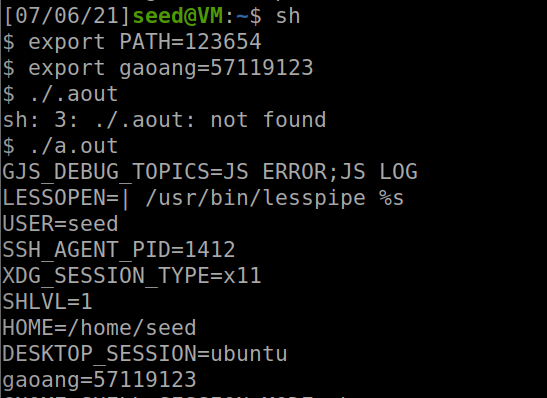


Task5

生成a.out程序（打印环境变量）



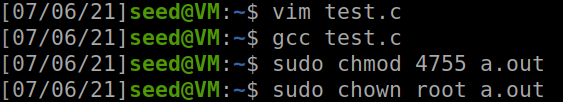
在shell中修改环境变量



发现更改的环境变量依然存在

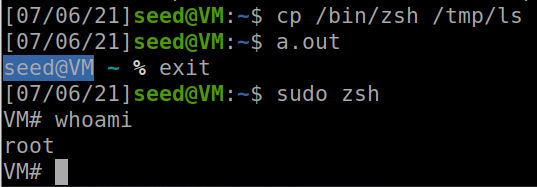
Task6

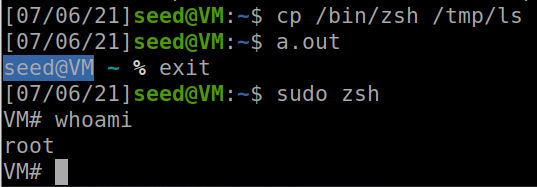
首先生成程序a.out，并使其变成setuid程序





将程序用拷贝指令到tmp目录下并命名为ls，并将环境变量设为tmp

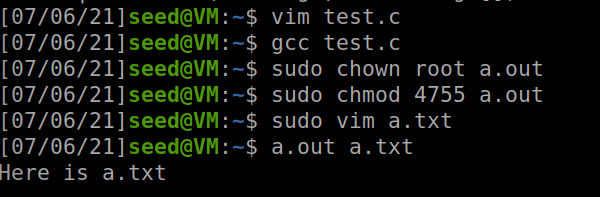




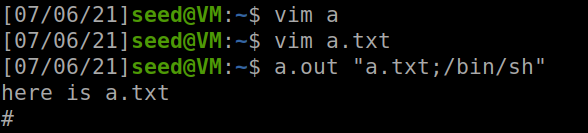
可以看到setuid程序打开了子程序，获得root权限。

Task8

首先建立一个root权限读写的a.txt，这时可以用程序读取a.txt

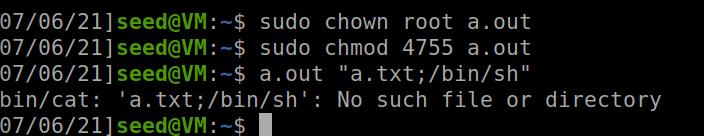


将输入改为加；的后缀时；运行程序时就会先调用cat指令，就可以调用一些root权限例如



尝试调用sh，成功，有root权限。

改为execve指令，发现execve将“a.txt;……”当成了一个文件名而不是两段指令。



结论：system指令会按照‘;‘对指令逐个执行，但对指令会有所保护，不会使其获得root权限，execve会将“”中的所有参数当作一个文件路径，因此不能逐个执行命令。