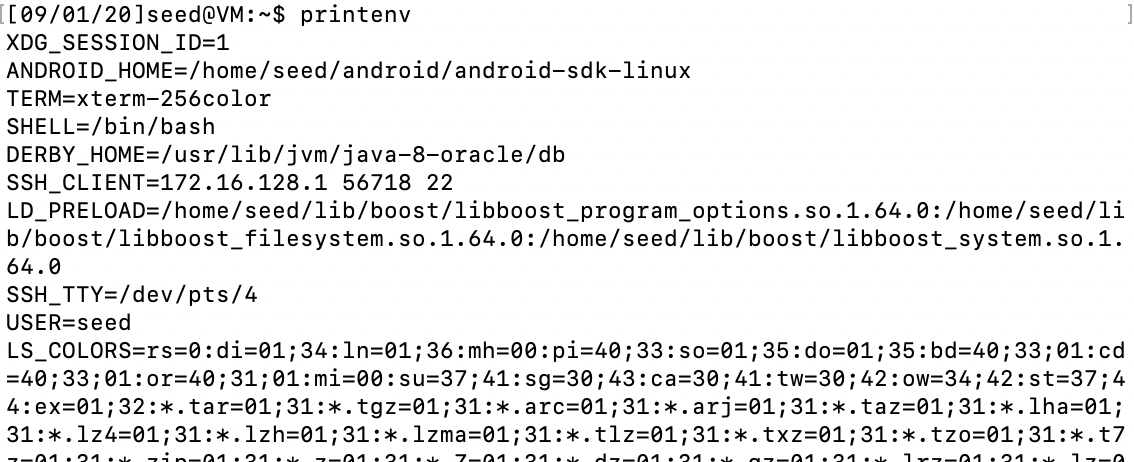
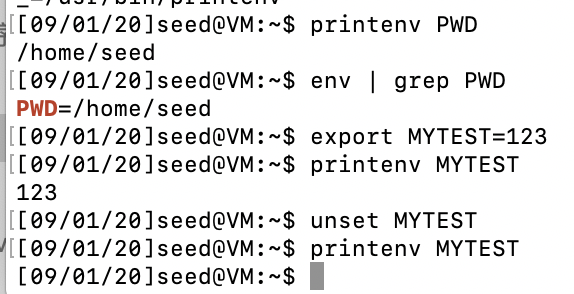
**Experimental Report**

**57117139**

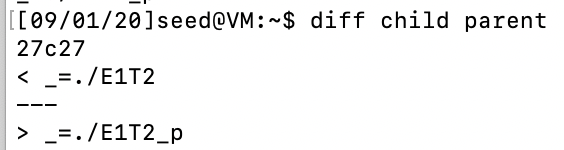
**Task 1**





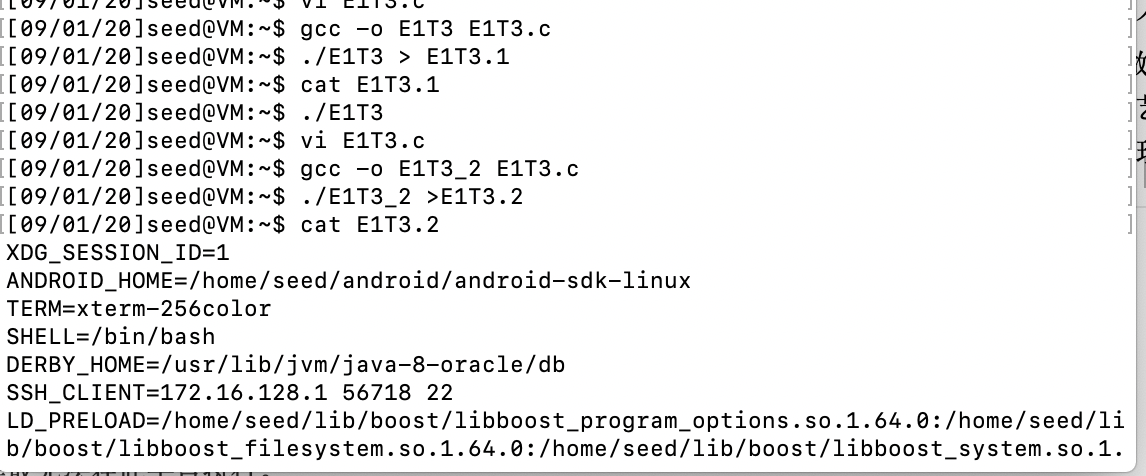
查看和修改环境变量

**Task 2**



子进程继承父进程的环境变量 所以两个程序的结果仅有文件名区分。

**Task 3**



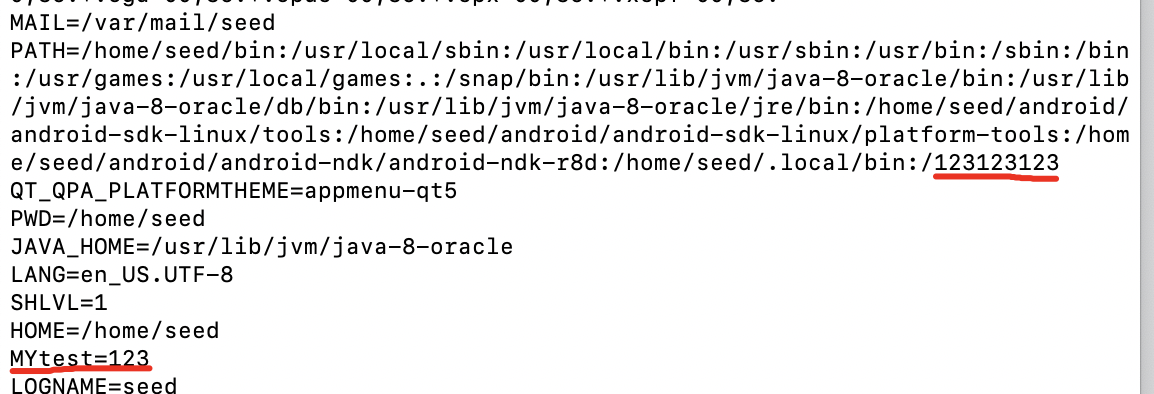
在执行Step1中的程序时无返回，说明没有自动继承，在执行Step2中的程序时返回环境变量，说明需要传递环境变量。

**Task 4**



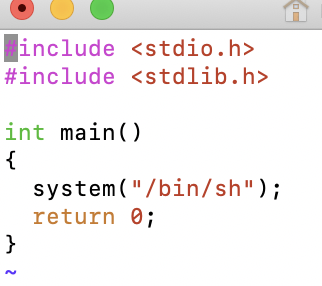
环境变量中有E1T4的名称，说明环境变量被传给了新的shell

**Task 5**

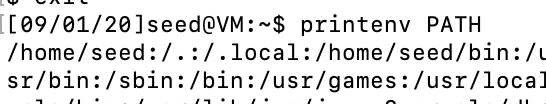


PATH和设置的新环境变量在child process中可以找到，但是LD\_LIBRARY\_PATH在child process中找不到。

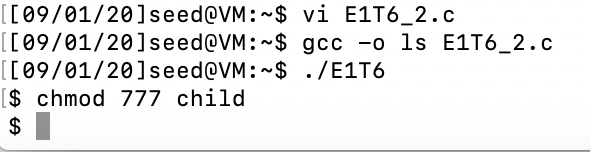
**Task 6**



自己写的ls如上图，编译后的程序改为root用户拥有 4755权限的setuid程序



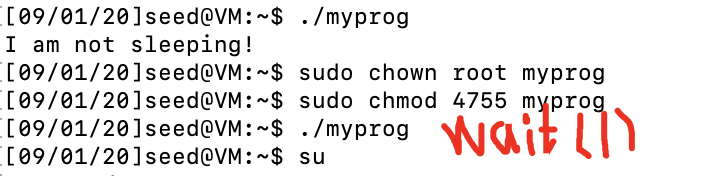
修改PATH的值，将/home/seed为第一优先级

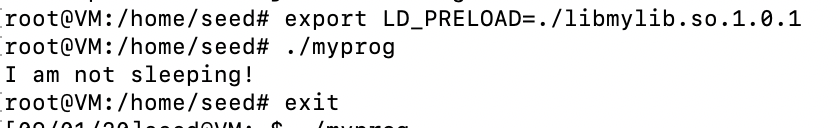


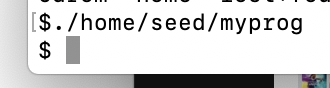
执行程序，程序调用ls时调用了自己写的ls打开了一个shell并具有root权限

实验说明 通过修改环境变量可以获得root权限

**Task 7**







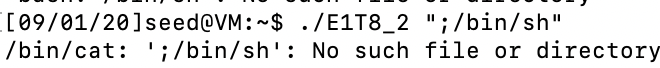
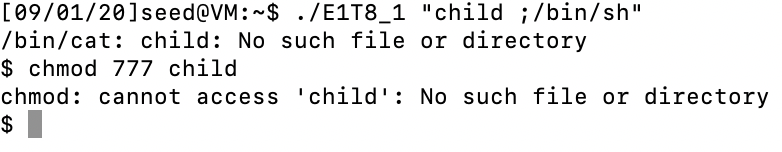
在Step2中 正常运行得到结果时 调用自建Sleep

Set-UID root program 在seed用户中运行，调用系统的Sleep

Set-UID root program 在root用户中运行，调用自建的Sleep

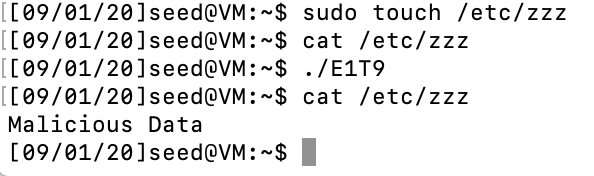
Set-UID uesr1program 在seed用户中运行，调用系统的Sleep

**Task 8**



使用system函数时，在参数中加入;/bin/sh将会进入root权限的shell中，可以对文件进行修改。使用execve函数时，使用相同方法无法达到相同效果。

**Task 9**



zzz文件被修改了，因为子进程调用的fd资源没有被关闭，依然可以实现root权限的读改zzz文件。