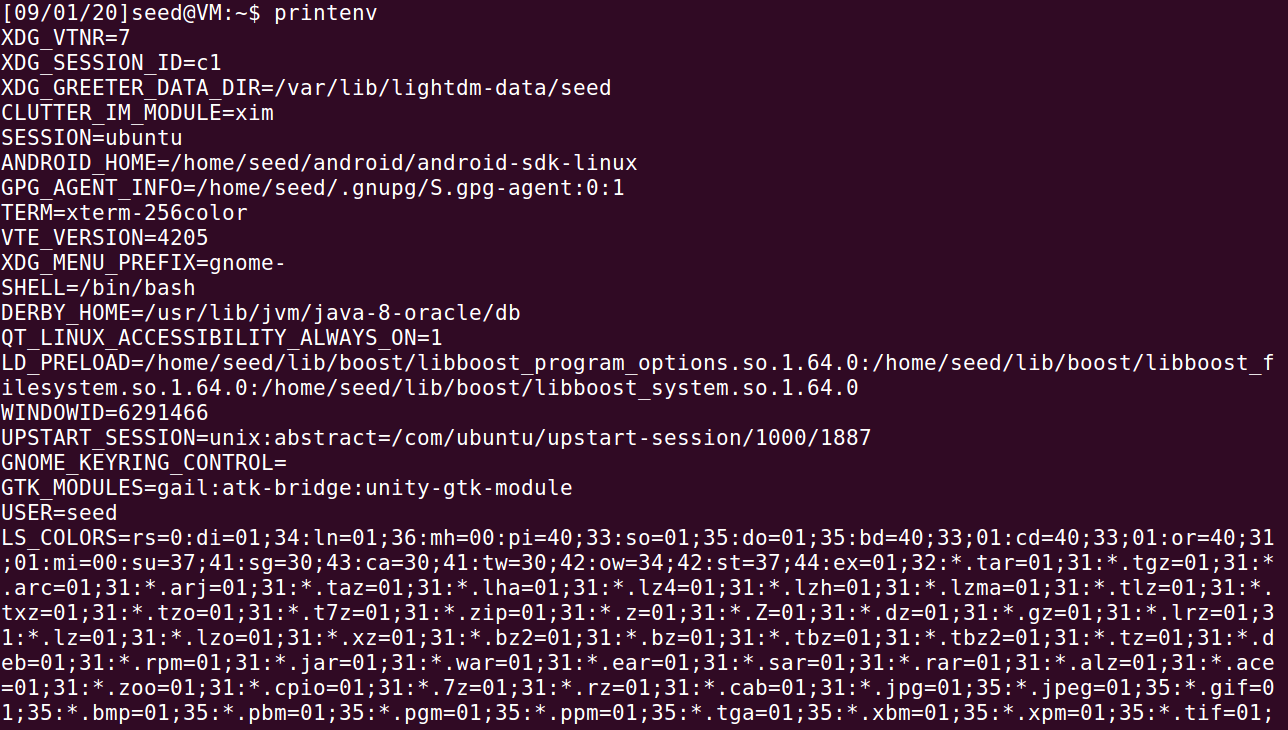
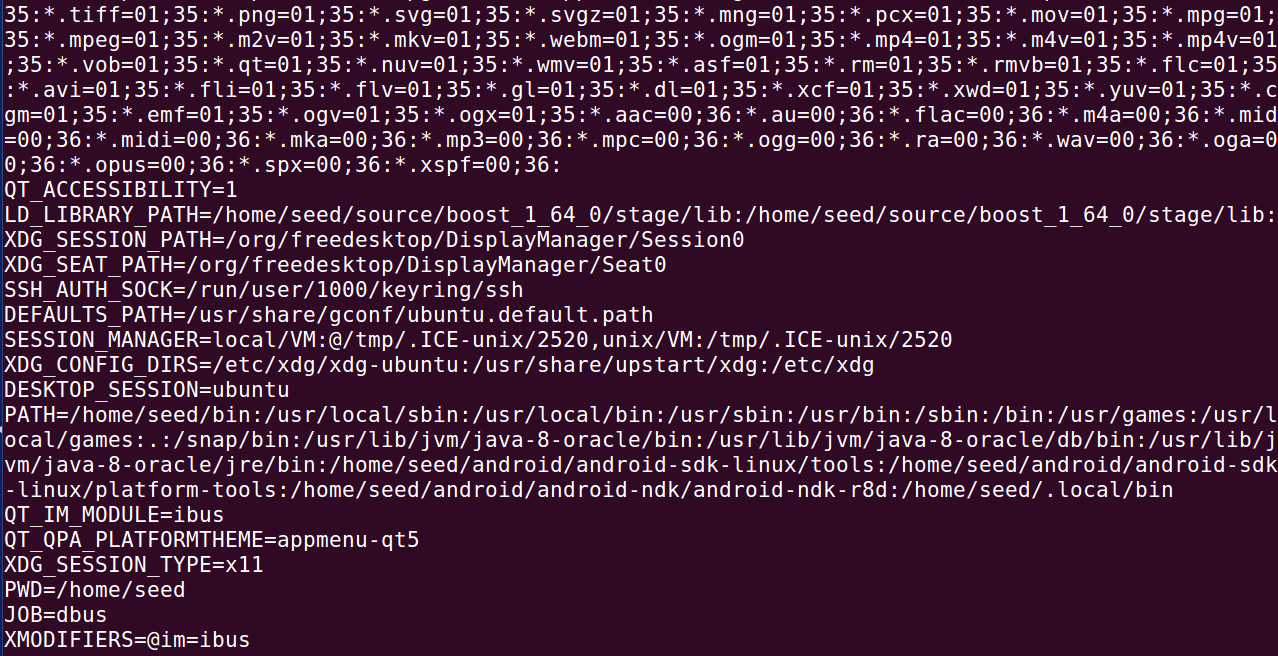
**Lab1**

57117208 傅靖邦

task1

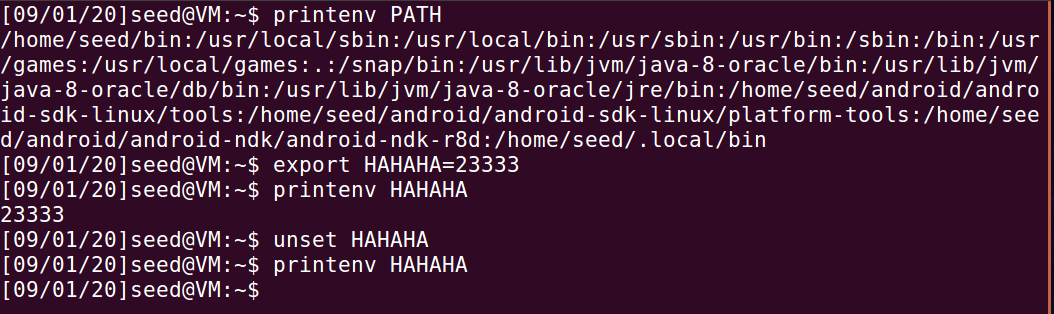
使用printenv打印环境变量







使用export unset创建、删除环境变量



task2

过程：代码复制到虚拟机中，命名为test2.c

使用gcc -o test2\_1 test2.c 命令编译

使用test2\_1 > child 命令把可执行文件的输出结果保存到child文件里

修改test2.c 打印父进程的环境变量，编译得到test2\_2

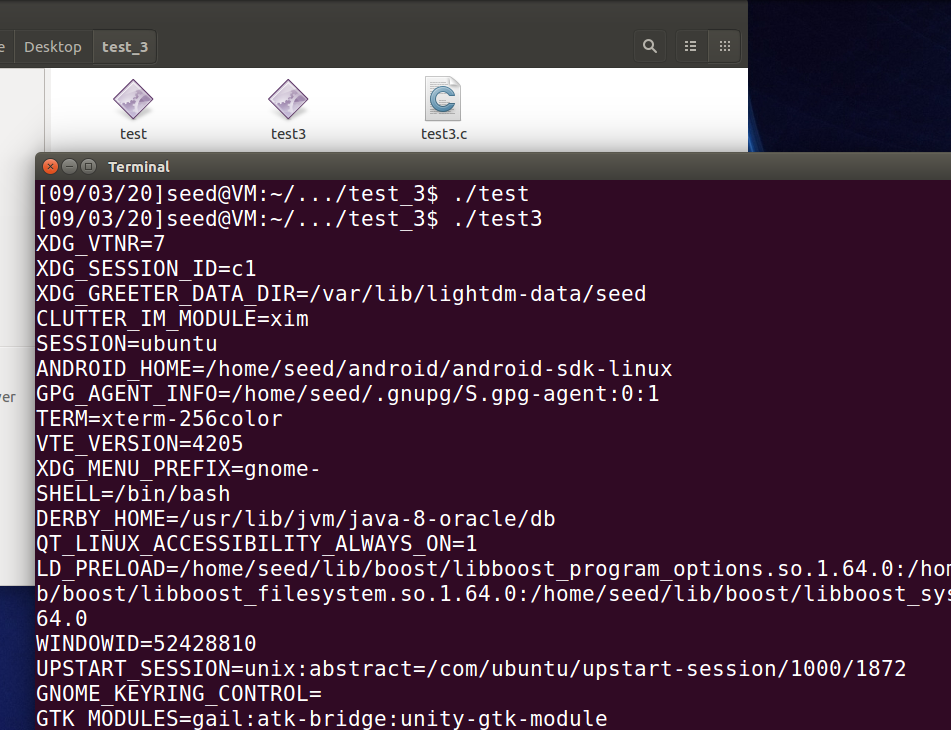
使用test2\_2 > parent 命令把可执行文件的输出结果保存到parent文件里

使用diff命令显示两个文件的区别



可以看出，两个输出文件的内容除了来源的文件名，没有本质上的区别。因此可以发现，父子进程的环境变量是相同的，即fork()会把环境变量完全继承到子进程中。

task3

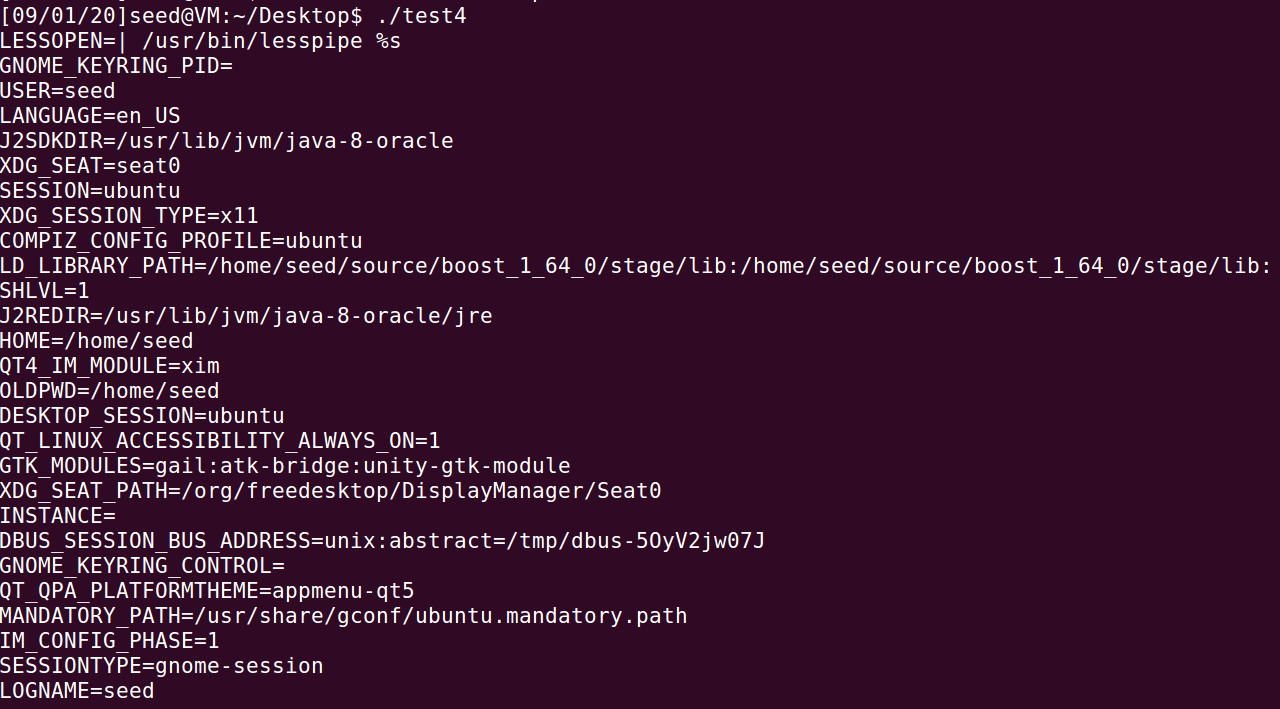


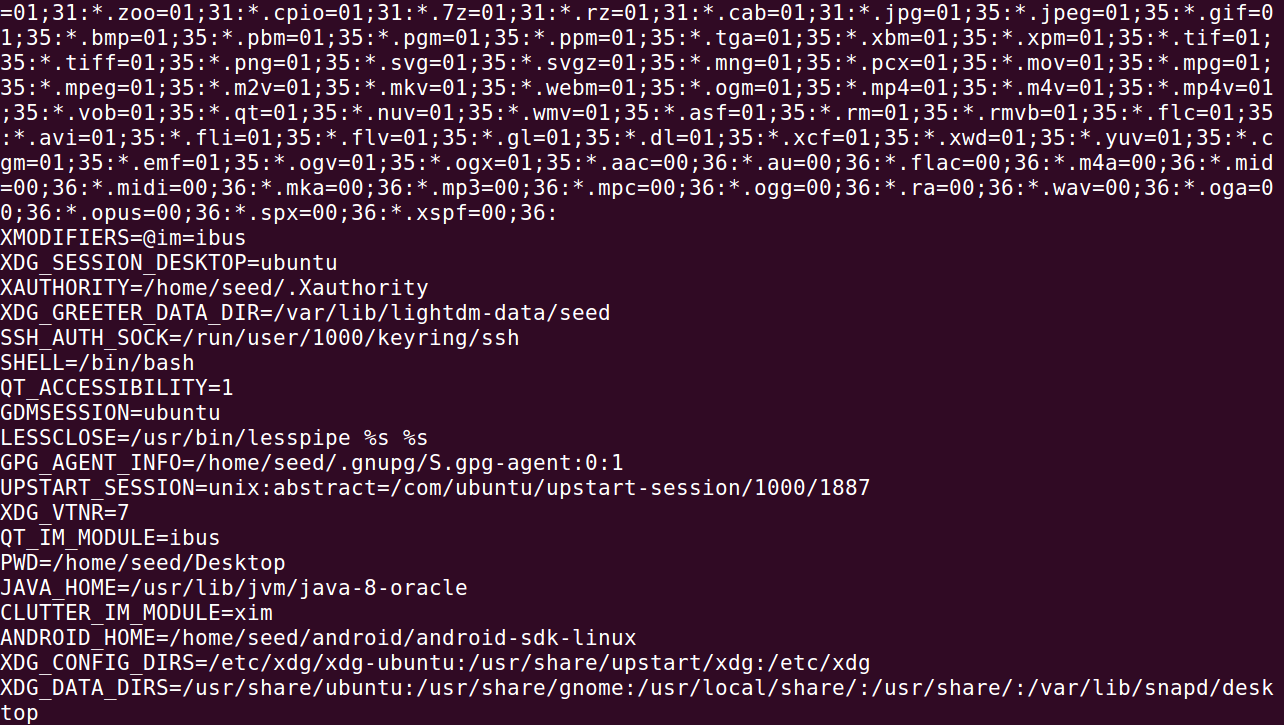
execve()的第三个参数为null时没有输出，修改参数为environ后会输出环境变量。execve()的第一个参数为执行文件的路径，本实验中为/usr/bin/env即输出环境变量，第二个参数是数组指针，用来给执行文件传参数。第三个参数为传递给文件的环境变量。

task4

编译并运行代码

运行结果

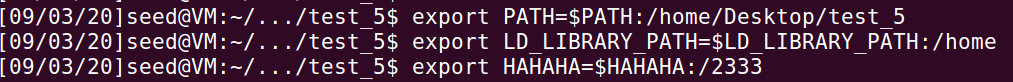


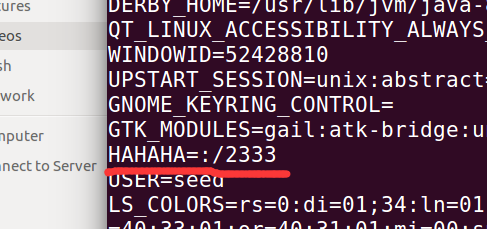


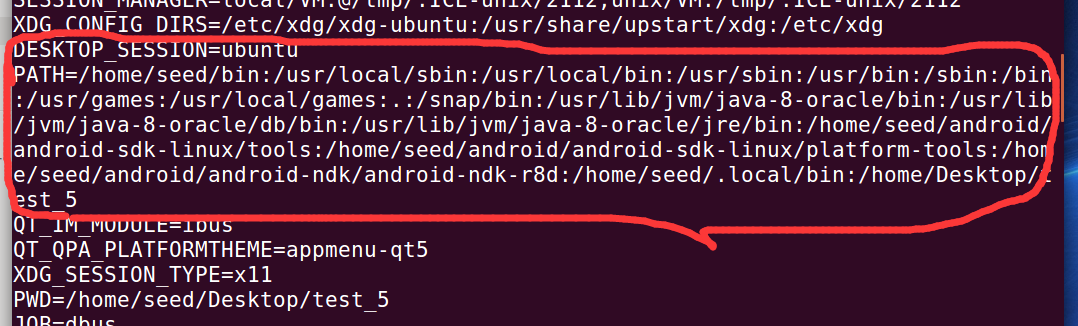
system()函数通过fork一个子进程，执行/bin/sh来指向一个shell，然后让这个shell来执行参数传递的命令。由于子进程继承父进程的环境变量，此处调用system()函数后会执行其中的命令/usr/bin/env，输出子进程和父进程的环境变量。（个人是这样理解的）

task5

编译能够打印当前进程所有环境变量的程序（代码在pdf中），把程序的所有者改成root,并转换为Set-UID程序。



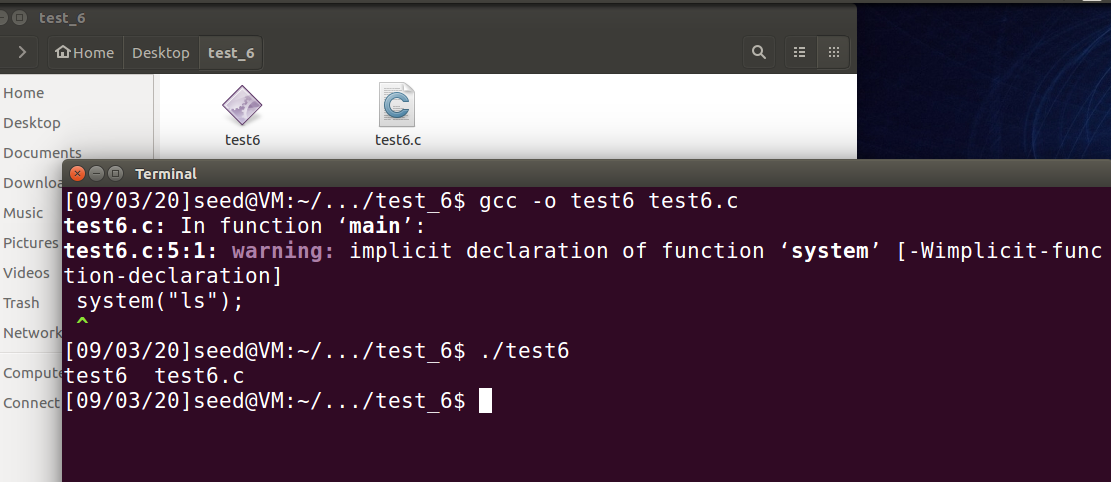




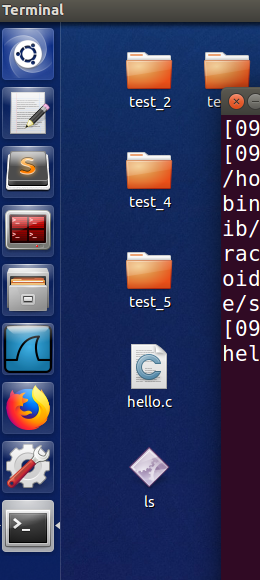
环境变量列表中没有找到新定义的LD\_LIBRARY\_PATH，但是新定义的HAHAHA和PATH都包含在了环境变量中。

task6

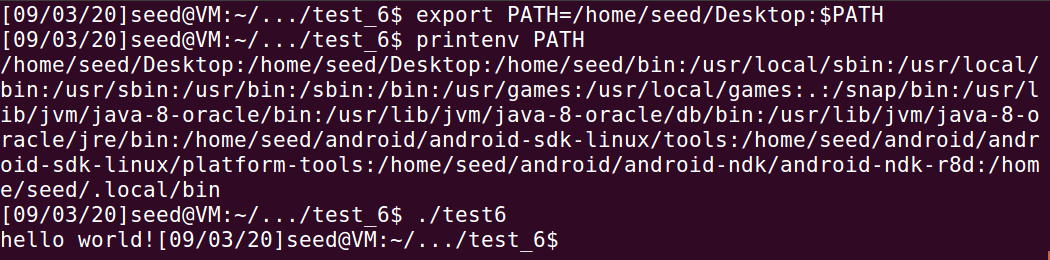
shell的行为可以被PATH等环境变量影响，用户可以自主设置环境变量，来控制Set\_UID程序的行为。在PATH的头部添加自定义的路径，并在该路径中放置名为ls的文件。运行时环境变量会进入自定义的路径中，运行相应的名为ls的文件。



在桌面创建自定义c语言文件，命名ls，输出hello world

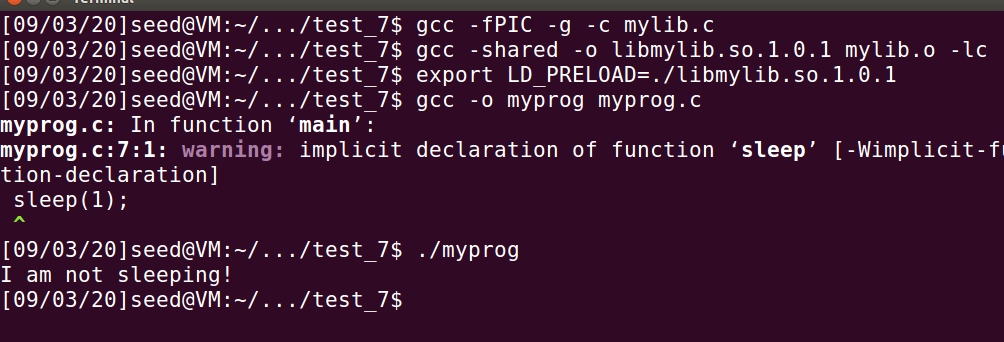


修改环境变量到桌面文件夹，再次运行test6文件。可以看出得到了自己设置的运行结果。

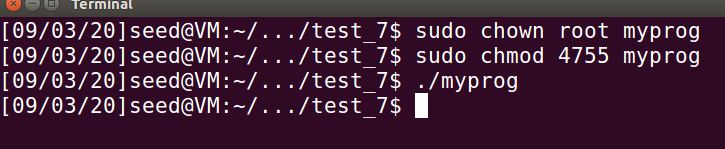


task7

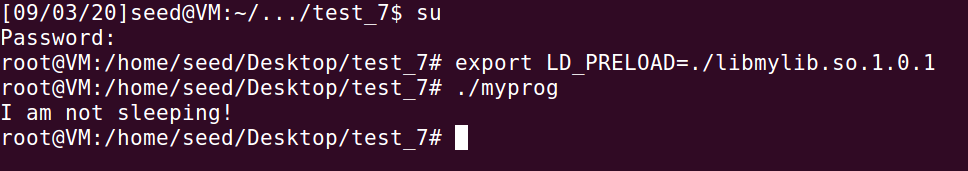
正常运行myprog:



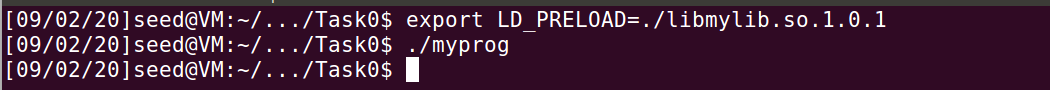
改为SetUID的root用户后运行，无输出



在root下修改环境变量并运行，有输出

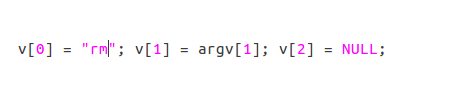


将myprog设定为user1的程序，在seed中exportLD\_PRELOAD环境变量并运行

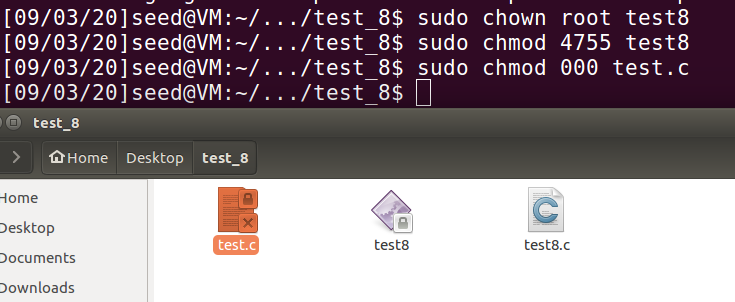
只有 ruid 和 euid 相同时， LD\_PRELOAD变量才会有效。当二者有差异时，该变量不发挥作用。

task8

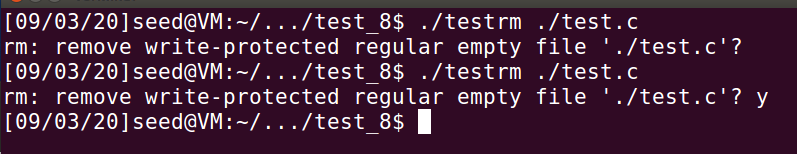
修改test8的代码 v[0]设置为删除指令，编译为testrm可执行文件



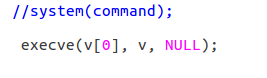
把test8的owner改为root，设为SUID程序。新建一个test.c文件，将其权限设为000.



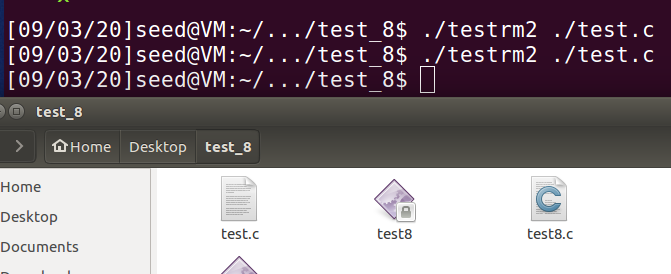
执行testrm



此时test.c被删除了。把system()注释掉，改为execve()，重新创建test.c，编译并运行

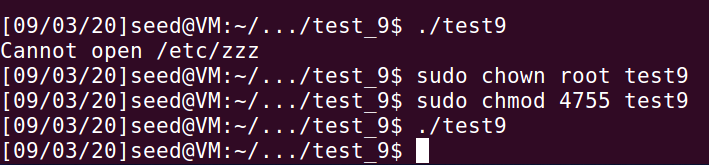


运行后发现test.c并未被删除。

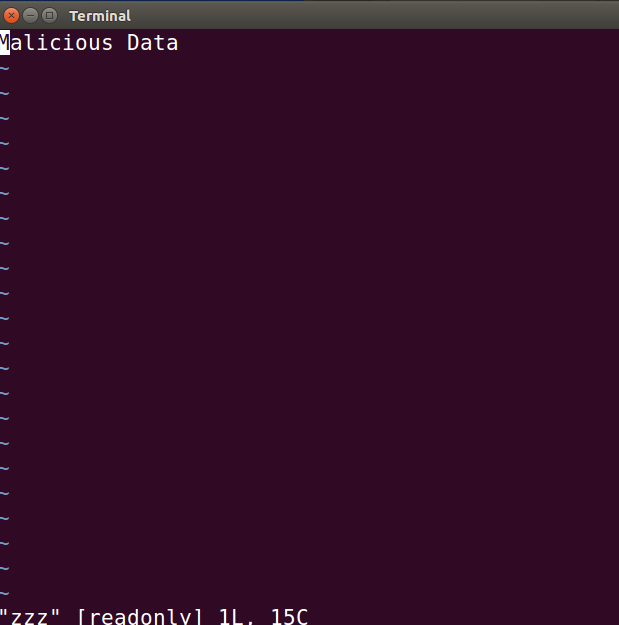


Execve 与 sysytem 方法的底层逻辑不同，system 方法本质上是开启新的shell 去执行一段 cmd，而 execve 则是传入可执行程序的参数的方式执行新的程序，execve 方法更加安全，System 方法在特权程序中被调用是不安全的。

task9



在给test9设置owner和SUID之前，test9没有权限打开文件。设置之后运行成功。



zzz文件被改写。

可见，test9运行过程中打开了zzz，撤销权限之前没有关闭zzz的文件描述符fd，此时fd还有root权限，可以执行写操作。